A probabilistic model for the evaluation of module extraction algorithms in complex biological networks

James Peter Gilbert, BSc

A thesis presented for the degree of

Doctor of Philosophy

UNITED KINGDOM · CHINA · MALAYSIA

School of Computer Science

University of Nottingham

United Kingdom

2015

Acknowledgements

I would like to thank my supervisors Jamie Twycross, Andrzej Bargeila and Andrew Wood for constantly providing support throughout my PhD, especially though difficult periods where the light at the end of the tunnel felt very dim. To Michael Holdsworth and Natalio Krasnogor, thank you for providing me with the opportunity and resources with this undertaking, making this work possible. To Paweł Widera, thank you for constantly challenging me during my first year of study and never being happy with work that was "good enough". To my family for the support they have given me over this period. I would like to all my friends and colleagues in Nottingham and Newcastle that helped the dull days go by and made sure the important things got done. But mostly I would like to thank Josie McCulloch for putting up with me for the last 3 years and being a constant source of support and inspiration.

Abstract

This thesis presents CiGRAM, a model of complex networks with known modular structure that is capable of generating realistic graph topology. Much of the recent focus on module detection has been geared towards developing new algorithms capable of detecting biologically significant clusters. However, evaluating clusterings detected by different methods shows that there is little topological agreement or consensus in terms of meta-data despite most methods discovering modules with significant ontology.

In this thesis an approach to modelling complex networks with ground-truth modular structure is presented. This approach is capable of generating graphs with heterogeneous degree distributions, high clustering coefficients and assortative degree correlations observed in real data but often ignored in existing benchmarks. Moreover, the model for modular structure concludes that non-modular random graphs are indistinguishable from modules.

This model can be tuned to fit many empirical biological and non-biological datasets through fitting target graph summary statistics. The ground-truth structure allows the evaluation of module extraction algorithms in a domain specific context. Furthermore, it was found that degree assortativity appears to negatively impact several module extraction methods such as the popular infomap and modularity maximisation methods. Results presented disagree with other benchmark models highlighting the potential for future research into improving existing methods in ways that challenge assumptions about the detectability of modules.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Introduction

The detection of modules, highly interconnected substructures that perform specific tasks, in complex biological networks is a considerable challenge that is of importance to many areas of biological hypothesis generation. The algorithms that perform these tasks are crucial to the development of our understanding of the inner workings of living things. This thesis concerns the development of tools to aid the analysis, evaluation and selection of module detection algorithms in a practical context. By providing realistic models that fit observed topological properties, such as heterogeneous connection distributions and highly transitive relationships, it is possible to provide an analysis of algorithmic performance. However, to date, current modelling approaches focus on general "universal" properties such as scale-free topology, rather than providing domain and context specific tests. The most significant contribution here presents an approach to generating random graphs with realistic properties such as heterogeneous connection counts and configurable correlation coefficients for connections between similar vertices. This introductory chapter aims to outline the motivation and aims of the thesis and gives a broad outline of each of the chapters contained within.

1.2 Background and motivation

Since the development of high throughput *"omics"* data collection methods, the biosciences have become deluged with big data problems that require the development of new methods of analysis [1]. Seeing the world through the eyes of a *"one gene one function"* perspective is a view of nature that is quickly being replaced with a view of systemic function. At the core of the methodology of this systems biology approach is the concept of a *biological network* [2]. This abstraction focuses on understanding the function of genes and proteins through their interactions with one another and the outside world.

This systems paradigm has given rise to the notion of biological *modules*; sub-networks of genes that perform specific, isolated functions that relate to testable hypotheses [3]. These modules have been shown to relate to known biological processes such as complexes within networks of protein interactions [4].

The detection of these modular components borrows heavily from the field of complex networks. This discipline focuses on uncovering how topology influences systemic behaviour [5]. The field has been popularised by notions such as *"small-worlds"* [6], where networks are characterised by short average path lengths due to properties such as *"scale-free"* [7] topology where extremely heterogeneous configurations are found to emerge in natural systems. These and related ideas have crossed over in to computational biology in many core areas.

The development of module discovery in biological networks is closely related to the idea of *community detection* in sociological networks [8]. Consequently, the terms *"community"* and *"module"* are used interchangeably throughout this thesis. This work promises interesting results, yet the recent explosion in the number of methods at the researcher's disposal [8] has created its own set of problems. Few benchmarks exist to evaluate community detection algorithms [9], and those that do exist are problem specific and lack the ability to properly mimic the topology of other real world networks.

1.3 Aims and objectives

The overall aim of this project is to develop a method to evaluate the performance of module extraction algorithms in the context of realistic topology. This requires the development of a modelling approach capable of generating ground-truth modular structures against which algorithms can be compared. In order to achieve this goal, a number of key points need to be achieved:

1. To evaluate current methods for validating clustering approaches through use of meta-data, highlighting any limitations.
2. Formally define what modular structure is and how it can be modelled.
3. Develop a model capable of generating synthetic complex networks with realistic topology and a known community structure.
4. Select the best parameters of this generative model in order to match the topology of real world datasets.
5. Evaluation of the impact real world topology has upon module extraction algorithms.
6. Development of a formal methodology for selecting appropriate module extraction algorithms in a domain specific setting.

1.4 Research questions

The above aims and objectives relate to several specific research questions to be asked in this thesis.

- **How do different module extraction algorithms compare to one another?** This question is of real interest to research in complex biological networks. If different algorithms produce different clusterings, it is important to understand the methods that can be used to aid selection.
- **How can a module be formally defined?** If one wishes to model networks with modular structure, a clear definition of what a community

actually is must be defined. In terms of computational modelling, a clear definition of modular structure that can be evaluated is required.

- **How can assortative structure be modelled?** Degree assortativity is an important topological property that is found in many real world networks relating to the propensity of nodes to connect to nodes of similar degree (defined more formally in Section 2.5.4) From the perspective of a probabilistic model, there must be an intuitive method of configuring the degree-degree correlations.

- **Can the developed probabilistic model be fitted to real networks or other specific topology?** Whilst a model capable of generating interesting topology is useful, it is only really an important tool if it can be tuned to fit empirical data. This requires an investigation into the distance measures and summary statistics that can be used to evaluated model fit.

- **Does assortativity impact the performance of module detection algorithms?** Degree assortativity is a feature observed in many networks that has not been widely modelled. This means that it is unknown as to whether a given community detection approach is impacted by correlated degree connectivity or not.

- **For a given network, which module detection algorithm is the best choice?** This question lies at the heart of this thesis. The wide array of module extraction approaches makes it difficult for researchers to select an appropriate algorithm for a given task. The use of accurate models with a known modular structure can aid in this decision, as well as helping with the improvement of algorithms for domain specific purposes.

1.5 Organisation of the thesis

This section outlines each chapter of this dissertation.

Chapter 2 provides the literature review for this thesis. This chapter first gives a broad overview of the graph theoretic definitions used throughout this

thesis. The remainder of the chapter can be thought of as being broadly broken into two sections. The first section consists of a review of relevant work with regards to biological correlation of expression, protein interaction and metabolic networks involved in this study. The second section of this chapter is concerned with the theory and topology of complex networks in a wider sense. Particular attention is paid to relevant models for the generation of topological structure as well as a review of existing module detection algorithms used throughout this study.

Chapter 3 then moves on to a core practical and theoretical application for the theory of complex networks in the form of whole genome correlation of expression data sets taken from plant biology. This serves as an evaluation of the state of the art in module detection. Particular focus is paid to the limitation of selecting such methods and ways to evaluate the detected communities using available, externally curated meta-data. An appendix to this chapter, Appendix A, also presents a web visualisation tool for these methods that offers bioscientists the ability to query the large scale datasets used in this study.

Chapter 4 introduces the **C**ircular **G**aussian **R**andom **G**raph **M**odel (CiGRAM). This is an approach to generating synthetic networks with realistic topology and community structure. CiGRAM is an extended form of fixed density random graphs that uses latent geometric variables to generate degree correlations and heterogeneity, with block structure to form modules. This approach makes the assumption that a module is indistinguishable from a random sub-graph, providing an approach for evaluating community detection algorithms.

In Chapter 5 the applicability of CiGRAM is validated. This comes in the form of evaluating the spectral properties CiGRAM is capable of generating, as well as the use of spectral distance and summary statistics to fit real world networks.

Chapter 6 formally demonstrates how CiGRAM can be used to evaluate community detection algorithms. One aspect of this work is the analysis of community detection algorithms in the context of assortative graphs, a property observed in the data sets evaluated in Chapter 3. The chapter then presents a formal methodology for the evaluation of community detection algorithms in the context of best-fit models from Chapter 5. This presents an approach

to algorithm evaluation and selection in a practical context. The chapter also includes a comparison of CiGRAM to other related benchmark graphs.

The thesis concludes in Chapter 7, where a summary of the contributions is provided. This also includes a discussion into how well the core aims and objectives of this work were met, as well as ideas for possible future directions of this research.

1.6 Contributions to knowledge

The research described in this thesis has demonstrated applicability of module detection algorithms to complex networks derived from correlation analysis of expression data sets. The use of statistical methods to aid biological discovery gives several specific biological hypotheses that can be experimentally validated, such as the relationship between co-expressed biological modules and evolutionarily conserved genes. This work also highlights a core limitation in the current methods due to the lack of agreement between the different module detection algorithms. This achieves one of the key objectives of the thesis; the evaluation of current approaches for validating detected clusters against known meta-data. A key finding is that the methods appear to be insufficient with regard to aiding algorithm selection.

The most significant contribution of this thesis is CiGRAM, which achieves the key objective of a model capable of generating a ground-truth modular structure. This model generates realistic modular structure through a simple assumption about the definition of a module, that it is indistinguishable from a random graph in terms of dividing into meaningful sub modules. This allowed the development of a methodology for the evaluation of module detection algorithms through the use of realistic synthetic models of datasets, another objective of the thesis. This methodology can be briefly outlined as follows and relates strongly to the structure of the thesis:

- Select a model capable of generating a known ground truth community structure and topology matching the real world network.

- Fit this model through optimisation or parametrisation to closely match

the empirical data.

- Generate multiple models with fixed levels of overlap and other parameters to provide a wide topological test bed.
- Test the algorithms on these models and select the best algorithm in terms of score.
- Validation of algorithms against available meta-data relating vertices to function.

Where multiple algorithms perform well, the additional meta-data step should be used evaluate clusters detected in real world data (such as the methods explored in Chapter 3) allowing the user to make an informed decision about algorithm selection. CiGRAM also allowed the discovery that certain key algorithms perform significantly worse in the presence of high levels of degree assortativity, a property observed in empirical data.

In addition to the definition of CiGRAM described in this thesis, Open Source software has been developed providing an extensible python library that can be used for module extraction evaluation. Appendix A also presents a set of web visualisations which provide the opportunity for researchers to explore the large scale expression data sets with a view to hypothesis generation. This is provided without the need to download and conduct a lengthy analysis of the data, as other sources of information are readily integrated into the tools.

Chapter 2

Literature Review

2.1 Introduction

The following Chapter reviews the literature related to the project. Firstly a preliminary section discussing the graph theory used throughout this thesis is provided. Next, the importance of modules in biological datasets is discussed, focusing on protein-protein interaction, metabolic and correlation of expression networks. The following section moves onto technical examples of how the global modular structure of networks is detected. Finally, methods for modelling the topology found in empirical datasets are discussed, giving a grounding for the benchmarking approach presented in Chapter 4.

2.2 Preliminary graph theory

This section gives the basic definitions of terminology relating to graph theory used throughout this thesis. The reader should note that the words *"network"* and *"graph"* are used interchangeably. A network can refer to any set of objects, referred to as nodes or vertices, that interact according to some specific pattern of edges. This notion inherently relates to the flow of information between objects in a system. If one imagines a particle taking a random walk around a network, stopping at each vertex, the set of vertices that can be visited is always dependent on the set of *adjacent edges* at the current vertex. A graph can be thought of as an intuitive map between related elements that could relate to direct interactions, correlations, or the notion of a discrete state space in which

each node describes the current state of some system and any adjacent vertices relate to the states that can be transitioned to. The concern of this thesis is the *topological* and structural properties underlying biological networks.

Beyond the definitions contained in this chapter, the terminology in this thesis is presented as and when the reader requires it. However, to aid quick reference, Table 2.1 displays common definitions with sections listed to aid comprehension.

2.2.1 Basic concepts

Formally, consider an *undirected*, *unweighted* graph G as a set of *vertices* V and *edges* E such that a pair of vertices i and j are considered to be connected if the tuple (i, j) is present in the set of edges E. By convention we will term the number of vertices in a graph as the cardinality of the vertex set $n = |V|$ and the number of edges as the cardinality of the edge set $m = |E|$. Simultaneously, we consider the *adjacency matrix* of a graph A to be the $n \times n$ binary matrix representing G such that $A_{ij} = 1$ if vertices i and j are adjacent and $A_{ij} = 0$ otherwise. Where a graph has edges that are non-equivalent, we consider this a weighted graph. In the case of a weighted graph, the elements of A can take on any real number.

In the case of directed graphs, or digraphs, we consider A to be non-symmetric and the order of (i, j) to be relevant. The direction of an edge indicates the available flow of information. If the link (i, j) is present within a digraph then information can flow from vertex i to vertex j, whilst if the vertex (j, i) is not present then no information can pass from j to i.

Node *degree* refers to the number of adjacent edges that a node has. We can consider the total degree of a node to be $k_i = \sum_{j \in V} A_{ij}$ or, in set theoretic notation, as the cardinality of the set of edges containing i, $k_i = |\{(i, j)|(i, j) \in E\}|$. The set of edges adjacent to a given vertex is its neighbourhood. In the case of directed graphs we can consider *in degree* as the cardinality of the set of adjacent edges pointing to the node, and *out degree* as the cardinality of the set of adjacent edges pointing to other vertices.

The density graph is an important property and should be considered when

comparing graphs in the presence of topological measures or summary statistics. We define the density of an undirected graph as,

$$d(G) = \frac{2m}{n(n-1)}. \tag{2.1}$$

When $d(G)$ is close to 1 the graph is considered dense, whereas graphs with density close to 0 are considered *sparse*. A similar calculation to density is the average node degree denoted by $\hat{k} = \frac{2m}{n}$. Note that a hard definition of *"sparseness"* depends upon the cardinality of the vertex set, in this case the values of the adjacency matrix are mostly zero. Almost all complex networks are considered sparse, and structural properties such as heterogeneity in the number of edges of each node often require sparse graphs [10].

A *subgraph* of a graph is any non-empty subset of the nodes and edges. A path, or walk, within a graph is any ordered sequence of vertices such that an edge exists between each vertex. The shortest path between two nodes is the path with the lowest cardinality; many such paths may exist. A graph is said to be *connected* if there exists a path between each pair of vertices. The largest connected component of a graph is the largest connected subgraph. An *induced subgraph* is a subgraph that contains a subset of verticies from V as well as all edges between them contained in the set E. An n-clique refers to any *fully connected* graph or subgraph that contains an edge between all pairs of vertices.

A cycle is any closed walk such that the path starts and ends on the same node, with no repetitions of vertices in between. A tree is a form of acyclic graph, that is to say that without cycles there is one and only one path between each pair of vertices. Trees are both necessarily connected and contain exactly $n-1$ edges, adding any edge to a tree will, therefore, introduce a cycle.

We define a *two star* as a path containing any triple of nodes, of specific interest is the number of two stars a node is central to. The number of two stars a given node is central to can be defined as,

$$s_i = k_i(k_i - 1). \tag{2.2}$$

This is an important property when considering local and global network statistics such as the clustering coefficient [6], described in detail later.

Another important measure relating to networks is the mean shortest path length. The mean shortest path length is given by,

$$l(G) = \frac{1}{n} \sum_{(i,j) \in V} sp(i,j), \tag{2.3}$$

where $sp(i,j)$ is the shortest path length between a pair of vertices. In Section 2.5.2, the notion of a small world network is discussed. In this context, the mean shortest path length is used to characterise a specific form of network.

2.2.2 Measures of topology

The following section summarises some of the important topological summary measurements that are used to characterise networks in this thesis. The reader is referred to a review [11] for a more comprehensive list of measurements.

Degree distributions

The degree distribution is an important summary statistic that will be discussed at length in this thesis. For simple graphs, a histogram is sufficient to model the degree distribution, selecting a number of bins appropriate to the network size. However, many of the networks studied in this thesis follow *heavy-tailed* distributions, making it difficult to select an appropriate number of bins [12]. Consequently, the convention of the *complementary cumulative distribution*, which considers the probability that you will find a node with degree greater than a given value $P(x \leq k)$, is adopted. These are viewed on a log-log scale in order to easily differentiate between distributions. Further information on heavy-tailed degree distributions is provided in Section 2.5.1.

Clustering coefficients

A triangle is defined as the triple (i,j,k), such that the three nodes form a complete subgraph, sometimes termed a *transitive closure*. An elegant way of measuring this *transitivity* is the *clustering coefficient* [6]. The clustering coefficient of a node is given by,

$$C_i = \frac{2t_i}{k_i(k_i - 1)} \tag{2.4}$$

where t_i is the number of triangles containing node i. Where $C_i = 1$ we can say that vertex i is in every possible triangle that it can be contained within, likewise where $k_i = 0$ we can see that i is contained within no triangles. This definition of the clustering coefficient is equivalent to the density of the induced subgraph of a node's neighbourhood. It is conventional to measure the mean clustering for a given network $C = \frac{1}{N}\sum_{i \in V} C_i$ against a random graph of equivalent density. When considering the clustering coefficient, the convention used throughout this thesis is to ignore nodes with a degree of 1 as they are not central to two stars.

Vertex criticality

The notion that a vertex is critical to the structure and function of a network is captured by *centrality* measures [11]. The most commonly used measure is betweenness centrality [13] which measures the fraction of shortest paths through a given node as follows,

$$B_u = \sum_{i,j \in V} \frac{\delta(i,u,j)}{\delta(i,j)}, \tag{2.5}$$

where $\delta(i,u,j)$ is the number of shortest paths between vertices i and j that pass through vertex u and $\delta(i,j)$ is the total number of shortest paths between i and j. The betweenness centrality calculation can also be applied to edges as Equation 2.5 allows u to be an edge or a vertex. This is an approach used, for example, in the Newman-Girvan algorithm for detecting modular structure [14].

In order to measure the dependence the graph has on a small number of vertices, the central point dominance of a graph is defined by,

$$CPD = \frac{1}{n-1}\sum_{i \in V} B_{max} - B_i, \tag{2.6}$$

Where B_{max} is the maximum betweenness over all vertices. CPD is necessarily in the range $[0,1]$, a value of 0 indicates that the graph is highly decentralised and not dependent upon any small number of vertices. In contrast, where CPD is close to 1, the network is highly dependent upon a small number of vertices with a high level of betweenness centrality.

2.2.3 Graph Laplacians

Spectral clustering is one of the oldest methods for partitioning graphs [15] and refers to methods that partition data according to using the eigenvectors of matrices. The approach taken with graphs is either to use the *Laplacian* or *Normalised Laplacian* of the graph rather than the adjacency matrix. The Laplacian of an adjacency matrix is given by,

$$L = D - A \tag{2.7}$$

where D is defined as the *degree matrix* of A, that is to say D is a diagonal matrix such that $D_{ii} = k_i$, the degree of each vertex. We can then see that the elements of L can be defined as,

$$L_{ij} = \begin{cases} k_i & \text{if } i = j, \\ -1 & \text{if } i \neq j \text{ and } A_{ij} = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

The normalised form of the Laplacian matrix is defined by,

$$\mathcal{L} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \tag{2.9}$$

where $D^{-\frac{1}{2}} = diag\{k_1^{-\frac{1}{2}}, k_2^{-\frac{1}{2}}, ..., k_n^{-\frac{1}{2}}\}$. From the definition in Equation 2.9 we can then see that the normalised Laplacian takes the form [16],

$$\mathcal{L}_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -\frac{1}{\sqrt{k_i k_j}} & \text{if } i \neq j \text{ and } A_{ij} = 1, \\ 0 & \text{otherwise .} \end{cases} \tag{2.10}$$

Conventionally, spectral methods use the eigenvectors of the graph as a projection of the graph onto a metric space. This allows one to use conventional clustering approaches such as k-means clustering to partition the graph space. However, many of the approaches used within this domain do not lend themselves well to complex, heterogeneous networks of the form we are interested in as they often assume properties such as roughly uniform cluster size [17, 18].

Whilst conventional spectral clustering is extremely limited in its application to large, heterogeneous sparse graphs found here, Fay et al. [16] summarised

the potential as a form of graph distance measure using the eigenvalues of the normalised Laplacian to characterise the structure of graphs. This approach is of little help to finding modular structure directly, but may have implications for finding appropriate models both with and without community structure. In Chapter 5 these distances are applied to fitting graphs models to empirical data.

2.3 Modules in biological networks

This section discusses the practical aspects of module discovery in biology, giving the reader some insight into the importance of modular structure. The idea of functional components is the basis of systems biology [3] and is at the heart of modern research. The extraction of meaningful biological functions promises to improve the understanding of living things through generating meaningful hypotheses about the co-regulation or common processes involved in biological systems. Integration of external data sources and multiple networks is a core aspect of modern bioinformatics [19], the focus of this thesis, however, is the discovery of structural modules from the topology alone.

This section is structured in terms of three classes of biological networks of interest to this study; gene correlation of expression (co-expression) networks, physical binary protein-protein interactions (PPI) and metabolic networks constructed from pathways of reactions. The limitations of these approaches cannot be understood without first understanding the objectives involved with these forms of study. This section reviews the methods of construction within these networks before discussing some exemplary studies into how modules have aided biological knowledge. This gives a grounding in the biological relevance of these modules, but is not a technical review of the function of the algorithms used. Following on from this section, Section 2.4 provides a review of the practical, computational approaches for finding modules in complex networks. This is, by no means, an exhaustive study of networks that apply to biological forms. Neural networks [20], Gene Regulatory networks [21] and many other forms of network are actively studied, and many of the methods discussed in this section have applications beyond those studied here [2].

Symbol	Description	Definition
G	Graph, collection of vertices and edges.	Section 2.2.1
V	Set of Vertices.	Section 2.2.1
E	Set of edges.	Section 2.2.1
m	Number of edges.	Section 2.2.1
n	Number of nodes.	Section 2.2.1
A	Adjacency matrix.	Section 2.2.1
A_{ij}	Binary variable for adjacency of two vertices.	Section 2.2.1
k_i	Degree of node i.	Section 2.2.1
$\hat{k}$	Average node degree.	Section 2.2.1
$d(G)$	Graph or subgraph density.	Section 2.2.1
$l(G)$	Mean shortest path length.	Section 2.2.1
C_i	Clustering coefficient of node i.	Equation 2.4
C	Network average clustering coefficient.	Section 2.4
B_u	Betweenness centrality of edge or vertex.	Equation 2.5
CPD	Central point dominance.	Equation 2.6
P	Partition of a graph.	Section 2.4.1
$\mathcal{C}$	Set of communities or modules.	Section 2.4.1
Q	Observed Modularity of a partition or graph.	Equation 2.11
Q_{max}	Maximal modularity partition of a graph.	Equation 2.11
r	Degree assortativity coefficient.	Equation 2.26
NMI	Normalised mutual information.	Section 3.3.1

Table 2.1: Definitions for symbols used throughout the thesis.

2.3.1 Protein-Protein interaction networks

If proteins are considered the mediator of biological action acting as part of complex signalling mechanisms, catalysing metabolic reactions and performing fundamental cellular processes like transcription, then their interactions surely determine much of the functional nature of biology. This whole scale, bottom up approach characterises a goal of systems biology and is often referred to as the *interactome* [22]. The objective of this line of study is no less than characterising every molecular interaction that occurs within an organism.

The scale of these networks is vast, for example, a recent study into the human interactome has experimentally collected 30,000 interactions between 14,000 proteins and this is only around half of the total interactions expected [23]. If one gene coded a single protein, the human interactome would contain over 20,000 proteins. Splice variants of DNA and potential post translational modifications greatly increase this number [24]. If one were to test this many interactions it would require 200 million protein pairs to be combined to generate a binary map.

As a consequence, the use of high throughput technology such as Yeast 2-Hybrid (Y2H) [25] and Tandem Affinity Purification (TAP) mass spectrometry [26] are required to generate large scale datasets. Y2H assays are based on modified yeast strains to indirectly measure the interaction of proteins. Given two proteins of interest protein x and protein y, x would be treated as bait protein and fused with the DNA-binding domain (BD) of a transcription factor, whilst y the "prey" protein is fused with the activating domain (AD) of the transcription factor. Only when the proteins interact is the AD in close enough proximity with the BD for a reporter gene to be expressed, thus giving a binary measure of the interaction between proteins.

A huge limitation of the protein interaction networks is that the collection methods are prone to error, reporting both false negatives and false positives. For example, most Y2H interactions are only published if verified by other sources [27]. In 2002 Von Mering *et al.* [28] concluded that the accuracy rate from high throughput experiments was as low as 20%. Whilst development in this regard has improved, Huang and Bader [29] still concluded that false

discovery rates and false detection rates are very high. The solution to this problem may lie with improved mechanisms for the collection of data, but until this occurs methods that model and validate mechanisms for biological network discovery are required. Despite limitations Y2H and TAP have been applied to map many of the protein-protein interactions of many organisms as highlighted by nearly 15 years of development [4, 30–32].

An alternative approach to building large scale interaction networks is to use online resources such as STRING [33] and BioGRID [34]. These databases store a wealth of literature-curated protein interactions and constantly grow in size. These resources also provide convenient programmatic APIs that allow the integration of datasets into reusable informatics tools as well as aiding the enrichment of gene sets and aiding error collection. However, TAP-MS and Y2H datasets have been shown to be far more reliable than literature curated networks. The work of Venkatesan *et al.* [35] highlights that methods that use approaches such as gene ontology should not assume that these attributes are free from bias. Indeed, in less well characterised organisms, much of the data regarding interactions appears to be highly erroneous, making it difficult to use for validating the TAP-MS and Y2H methods discussed above.

Modules in interaction networks

The manner in which proteins interact characterises the function of living things. The notable first work into detecting groups of functionally related proteins in complex networks is that of Spirin and Mirny [36]. It is natural to be drawn to the idea of a clear group of proteins that bind together at a specific point to form a molecular machine. For example, RNA splicing requires a number of proteins to act as a singular macro-molecule.

Alternatively, one can consider a cluster of proteins that form aspects of a functional process, these proteins interact but at different time scales. For example, protein kinases involved in signal transduction will interact with one another, but their expression will be controlled by environmental or metabolic processes. From the perspective of interacting groups in a binary map, the time delay is not encoded information and so both functional processes and complexes will be detectable in a similar manner. In the case of functional

modules, protein interaction networks offer insight into the function of biological processes that may take place over long time scales making it difficult to directly generate hypotheses from conventional lab based methods. By detecting these groups in binary interaction maps, hypotheses about functions over extremely long time scales can be generated, highlighting the potential of module detection approaches.

The authors of [36] developed a method for uncovering modules based on the betweenness centrality of proteins which proved effective at the time. However, there has been extensive development in both algorithms for detecting modules [8] and datasets [4, 31, 32]. As a consequence, hierarchical and overlapping approaches have been developed and applied in more recent studies [37].

The work of the Arabidopsis Interactome Mapping Consortium produced an accurate interaction network which was developed with Y2H [4]. To date, very little further work has been conducted into plant protein interaction networks due to the difficulty of applying TAP-MS [38], making the Arabidopsis interactome a valuable source of information for plant biology. In [4], edge clustering [37] was applied and discovered a number of experimentally verified modules such as those related with signalling pathways in barley [39].

2.3.2 Correlation of expression networks

Microarrays output a gene expression profile for tissue under some experimental condition. Many gene or whole genome analysis is possible depending on the specific probe sets available [40]. Microarrays work by laying down a number of probes that match RNA or DNA sequences through hybridisation; i.e. a probe is a specific complementary sequence of DNA or RNA that matches all or part of a gene transcript. Each probe will be within a certain area and the number of molecules that a probe matches can be counted (e.g. through florescent dye produced as a result of hybridisation). Design is a key issue with these experiments as data can often be noisy or prone to sample bias and, as a result, appropriate statistical procedures need to be developed before an experiment to give a normalised view of expression levels [40].

Whilst microarray experiments are still a popular source for the collection

of gene expression data, they are fast being replaced by the use of high accuracy RNA sequencing (RNA-seq) [41] techniques that do not require a whole genome to be sequenced, eliminate bias associated with probes and offer improved accuracy. Rather than matching genes through hybridisation, which often introduces bias to experiments, RNA-seq matches sequences, meaning a fully sequenced genome is not required. Fundamentally, however, both technologies measure tissue and time specific genome wide expression levels [42].

Due to the developments in transcriptomics that allow the analysis of the entire transcriptome, methods such as correlation of expression networks are a popular method for the analysis of datasets. Here, gene expression levels are measured at either different time points, or in different environmental conditions. A correlation matrix is generated based on the expression profiles of each gene [22]. This can be converted into a graph by selecting a correlation score threshold which can nominally be a specific level of confidence. For example, the correlation threshold for which 95% of potential interactions are excluded ($p < 0.05$). An alternative method for selection could be basing it on some topological feature that is wished to be observed, for example, in the work of Bassel *et al.* [43] the threshold was selected because this maximized a power law distribution for the network. Alternatively, this could be a rank of the top N co-expressed genes or a threshold based on particular genes of interest [44]. The selection of the correct correlation threshold for a given network is a balancing act between removing spurious edges that limit the analysis of data and maintaining enough information such that some network inference can be conducted.

It is important to point out that, where correlation networks are concerned, an edge does not indicate a direct interaction between genes. To establish a causal link one must provide evidence in support. Correlation networks work under the *"guilt by association"* principle; when genes are expressed in the same tissue at the same time across multiple samples in response to similar stimuli it is likely that they are related [44].

The study of SeedNet by Bassel *et al.* [43] highlights the potential to use genome wide expression data to elucidate biological function. Here, the authors collected multiple microarray experiments of publicly available data

from different sources in order to investigate seed germination. A co-expression network was then created and clusters were generated through agglomerative hierarchical clustering of the microarray data. Combining previous experimental data showed a clear relationship between clusters and genes known to be associated with germination and non-germination. This use of known data should be considered exemplary in the analysis of this form of data as it generates the hypothesis that unknown genes within associated clusters are likely to influence or be influenced by the same regulatory processes.

The idea of a module in a correlation of expression network relates to the notion of *co-regulation* and relates strongly to the notion of "guilt by association". The general formulation is that, if a group of genes have a similar expression pattern over a time course or set of experiments, they should form a dense cluster. These clusters then form a hypothesis that the genes contained within are regulated by the same transcription factor or transcription factors. In Chapter 3 the analysis of topological clusters is conducted on SeedNet, two further Arabidopsis datasets and a Tomato Fruit Ripening Network. This includes combining external data sources, such as gene knockout experiments, to aid the understanding of these clustered groups.

2.3.3 Metabolic Networks

Metabolic networks are crucial to the understanding of biological systems. At any given time, a huge number of metabolic interactions occur within living cells, this can be characterised by the transformation of metabolites into substances that are useful to biological organisms, normally catalysed through the use of enzymes. One can represent these networks as directed networks between reactants and products [45] or, in a similar vein to the correlation of expression networks described above, the relative level of expression of metabolites at given time points can be used to form an undirected edge under the "guilt by association" principle [46]. Databases such as KEGG [47], WikiPathways [48], EcoCyc [49] and MetaCyc [50] store various amounts of metabolic interaction data for organisms including low level pathways and full organism metabolic maps. In this work, metabolic networks are treated as sets of metabolites that

share an edge if they are linked by a reaction. Extremely common so called "currency" metabolites, such as ADP or H_20 are generally removed. This is a common approach taken that sacrifices much of the complexity of the system in order to simplify analysis [22].

A major work in the analysis of large scale metabolic networks came at the turn of the century with the Work of Jeong *et al.* [51]. Here the authors discussed the "scale-free" nature of the networks, referring to the extreme heterogeneity of node degree (a topic discussed in Section 2.5.1), though little attention was paid to the specific modular structure. Later, work by Ravasz *et al.* [52] was conducted into the hierarchical organisation of metabolic networks by proposing a model in which networks form dense modular structures. Many of theses detected structures correlate strongly with known groups of pathways, indicating the value of uncovering modular structure.

Flux balance analysis [53] is a widely used tool that uses metabolic network models to calculate steady-states of biological systems, with respect to metabolites present at any given time. However, more precise analysis of enzyme kinetics is often limited when dealing with large scale complex networks and requires breaking networks down into smaller sub components that can be modelled and experimentally validated. This can be achieved through first hand expert knowledge, however, as datasets grow in size the combinatorial explosion makes automating the discovery of meaningful modular components a necessary step for this form of analysis [22].

The work of Guimera and Amaral [54] highlights one of the best early examples of applying module detection algorithms to complex metabolic networks. By applying modularity maximisation to 12 metabolic networks from different species, the authors were able to create what can be described as a *functional cartography*. The role of nodes within the network can be associated with inter-modular communication or intra-module function. This is done by computing two measures, the within module degree and the participation coefficient.

Comparing nodes by their level of participation within a modular structure allows the creation of several broad groups; nodes that are peripheral and contain all or most of their edges inside their own module, inter-module connectors

that are either hubs or non-hubs, "provincial" hubs that have most of their edges within their own module, and "kinless" hubs and non-hubs that have connections largely between communities. A common problem with much of the historical analysis of metabolic networks has been the limitation of modularity based algorithms [14] that do not allow overlapping vertices. More recent methods of community detection have moved into the discovery of overlapping modules [37], which is a distinctly different problem. Further details on specific methods for module detection are discussed in Section 2.4.

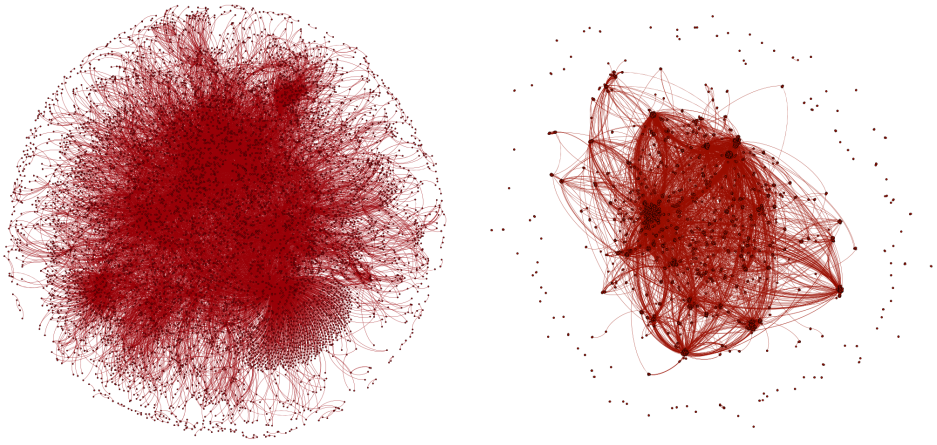
Aside from the numerical analysis, the annotation of functional modules through mapping them to KEGG pathways [47] shows a good example of how clustering, combined with visualisation, can be used to relay information from computational studies to domain experts. The use of modules as a form of visual "map" is a topic that merits further discussion and is explored in the next section.

2.3.4 Visualisation of biological networks

The need to make sense of large scale networks is of great interest to systems biology research [55]. The objective of network visualisation tools should be to present complex data in an intuitive fashion that allows interpretation. Specifically, visualisations need to focus on hypothesis generation, aiding experimental design [56].

The conventional "Hairball" forms of force direct layouts [57] are often found in publications. This way of viewing data does not help knowledge discovery. Alternative approaches of visualising data attempt a cluster based approach. For example, the OpenOrd layout [58] improves on conventional force directed layouts and ignores edges over larger distances. Figure 2.1 contrasts these two methods highlighting the difference between these approaches.

However, an aesthetically pleasing visualisation is only useful if it conveys meaningful information. For this to occur, the network visualisation must be based on a meaningful set of clusters and integrate external sources of information. Numerous software packages exist with regards to integrating visual information, such as Pajek [59] and ONDEX [60]. Perhaps the most



(a) Fruchterman-Reingold *(b) OpenOrd*

Figure 2.1: Examples of Fruchterman-Reingold and OpenOrd "hairball" visualisations of the Arabidopsis Protein-Protein interaction network taken from BioGRID [34].

popular tool of all is Cytoscape [61]. Cytoscape has the appealing aspect that a huge number of plugins are available [62] which allows integration of other sources of information such as gene ontology [63]. ONDEX [60], is an interesting approach to integrating different sources of information. The objective here is to combine experimental data from a number of sources such as KEGG [47], Transpath [64] and AraCyc [65]. Combing external information not only aids interpretation of datasets, but crucially, allows the generation of hypotheses.

In terms of displaying clusters, many approaches to the visualisation involve grouping nodes together by common shared attributes or topology. For example, the *clusterMaker* [66] Cytoscape plug-in allows users to cluster genes into relevant groups. This approach makes annotation easier and allows the researcher to investigate large scale, macro level interactions between functional groups.

In Chapter 3, a web visualisation of a Tomato fruit correlation of expression network is presented that uses the clustering algorithm found within the OSLOM module detection tool [67] in order to group related vertices. Whilst not so much a limitation of this approach as with the underlying clustering process, grouping nodes together requires one to trust the accuracy of module detection algorithms. The following sections go into more detail with regards to these limitations, but it must be stated that any visualisation that uses this approach

should be treated with a healthy degree of scepticism.

2.3.5 Discussion of biological modules

It is clear that the idea of detecting clusters within networks is an important goal within systems biology. This has been the focus of understanding biological systems, made up of thousands of genes and potentially millions of interaction, for around over a decade [3]. In this section, we have looked at some exemplary studies that have focused on the functional biological modules in protein interaction and metabolic networks as well as strongly co-expressed genes that aid data analysis. Modules also offer an interesting approach to visualising large scale data, offering a way to tame the unruly "hairball". Much of this work, however, relies on experimental validation and combining other sources of information such as gene ontology. Whilst often extensive, the granularity and inaccuracy of information appears to be a problem, computational approaches offer opportunities to solve these issues [68]. In Chapter 3, it is shown that when moving from a well studied organism like *Arabidopsis* to a less well researched one like the Tomato, the gene ontology becomes significantly less reliable. Different approaches will likely give different results and more focus on understanding the assumptions relating to the formation and detectability needs to be considered.

There are several competing hypotheses for the origins of modular organisation within protein interaction and metabolic networks. One such hypothesis is that the modular organisation is a result of natural selection optimising for the minimum number of links [69]. The argument here is that maintaining redundant links is not beneficial and, gradually, modular structure emerges due to its efficiency. Gene duplication models [70, 71] offer the alternative hypothesis that biological systems evolve by a process of copying, with these models appearing to create higher modularity than one would find by chance [72]. This hypothesis argues the modules are simply a by product of the process of gene duplication rather than modularity being specifically selected for. However, recent work has found that the existing gene duplication models may not be the best representation of real world protein interaction networks [73].

Whilst there is debate in the literature about the underlying cause for modularity in networks, there is clear evidence for modular structure in protein interaction and metabolic networks [52, 72, 74], and these modules are known to relate to functions. Uncovering these modules is a problem in computational biology and determining modular function is a vital aspect of systems biology. Detecting modules in networks is a purely hypothetical act and experimental validation must be completed in order to test hypotheses, a process vastly more time consuming than computational analysis. Section 2.4 now moves on to the technical aspect of detecting communities in large, complex networks.

2.4 Methods for module extraction

Despite the volume of literature and the number of algorithms related to the subject, there is still no widely agreed upon definition of what a module or community is [8]. The most commonly adopted assumption is that a community is a group of nodes that is more densely connected internally than externally. Thas has led to the definition of both overlapping and non-overlapping network structures. Approaches such as Infomap [75] and Modularity maximisation [14] find a single module for each vertex, whilst clique percolation [76] and Link communities by Ahn *et al.* [37], cluster vertices into more than a single group. In this section, we briefly discuss the competing definitions for modular structure before reviewing some of the popular methods for uncovering modules in large scale complex networks.

2.4.1 Covers and Partitions

There are two formal definitions of the block structure in networks, *partitions* and *covers*. A partition is a set of clusters or communities on the vertex set $P(V) = \{c_1, c_2, ..., c_n\}$ such that each node is contained within one and only one community. Each community has the condition that it must be an internally connected induced subgraph of G. An equivalent definition of a partition is a *cut set* on the edge set E. In the definition of a partition, each edge is either inside an induced subgraph c, or lies between two such subgraphs. We then define a cut set as a proper subset of E, on the condition that for each cycle

within G if an edge within the cycle is inside the cut set, there must be no path between the nodes that it connects. In practical terms, this means that for every cycle either two or more edges are contained within the cut set or no edges are included. Whilst some focus has been paid to the attention of the relationship between cycles and modular structure [77], this is an an area that is certainly open to further exploration. Furthermore, there has been no formal explicit association between cut sets, cycles and partition based methods. Another implication for the use of cut sets in partition based methods is that it demonstrates that the search space for any objective function is no larger than 2^m possible partitions.

Covers refer to overlapping modular structures. A cover is, again, a set of clusters that form internally connected induced subgraphs of the parent graph, $C(V) = \{c_1, c_2, ...c_n\}$. The distinction between a partition and a cover is that a vertex can be the member of many modules in a cover based approach.

The above definition could be considered a *crisp* definition of a cover in which a node is either a member, or not, of a given cluster. We also encounter a so called *fuzzy* definition of a community in which a node has a degree of membership to a given cluster e.g. $\mu_c(i) \in [0, 1]$ such that $\sum_{c \in C(V)} \mu_c(i) = 1$ [78]. Figure 2.2 visually shows covers and partitions.

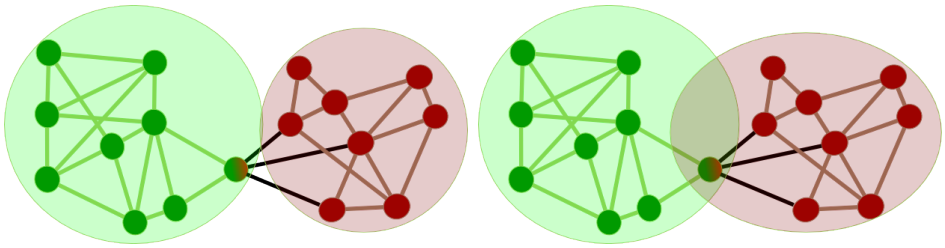


Figure 2.2: Example of a partition (left) and cover (right). The central overlapping node could be considered to be contained within two communities.

The remainder of this section discusses a number of methods for uncovering global modular structure in large graphs. This is broken into broad subsections: modularity based methods [14], information theoretic approaches [75], the OSLOM algorithm [67] and methods based on the propagation of messages in a simulation [79]. This is, by no means, a comprehensive study of the algorithms and approaches for module extraction. For a more detailed view, the reader is

referred to a recent review [8]. Instead, the following sections aim to give the reader a practical grounding in the methods used in Chapters 3 and 6.

2.4.2 Modularity

Newman and Girvan proposed a measure of modularity [14] for any given partition of a graph. Modularity can be seen as both the measure of the quality of any partition in the set of all possible partitions and for the overall modular structure of a graph. Here, the partition quality function, Q, is defined under the condition that a null model graph has no community structure.] The null model is based on the probability of two vertices forming an edge in a null model that preserves degree and assumes that there is no increased probability for subsets of vertices to form edges. More formally, under the null model two nodes i, j are assumed to connected with the probability $\frac{k_i k_j}{2m}$ where k_i is the degree of vertex i and $m = |E|$ the total number of edges in the graph. This is identical in form to the Chung Lu model described in Section 2.5.3. Formally, the quality function is given by

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \tag{2.11}$$

where the binary variable $A_{ij} = 1$ when there exists and edge (i, j) and 0 otherwise, c_i is the community in which i is placed inside and the function $\delta(u, v)$ is the Kronecker delta, $\delta(c_i, c_j) = 1$ if $c_i = c_j$ and 0 otherwise.

A more convenient form of equation 2.11 can be found in [80],

$$Q(P) = \sum_{c \in P} \left[\frac{m_c}{m} - \left(\frac{\sum_{i \in c} k_i}{2m} \right)^2 \right], \tag{2.12}$$

where $c \in P$ is the induced subgraph community within a given partition of a graph and m_c is the number of edges in c. We refer to Q_{max} as the maximal modularity observed in a graph which can be seen as a summary statistic of a network's topology. An edge only contributes to modularity if it is inside a community. Consequently, the modularity score will be higher for partitions with more edges inside communities than between them.

There are a huge number of heuristic based algorithms for finding the partition with maximal modularity. Approaches such as simulated annealing

[81], agglomerative methods [82] and genetic algorithms [83] have been tried. The work of Brandes *et al.* [80] discusses the difficulty of the problem and showed that it is NP-Complete, meaning that there is no polynomial time algorithm to maximise modularity in every instance unless $P = NP$.

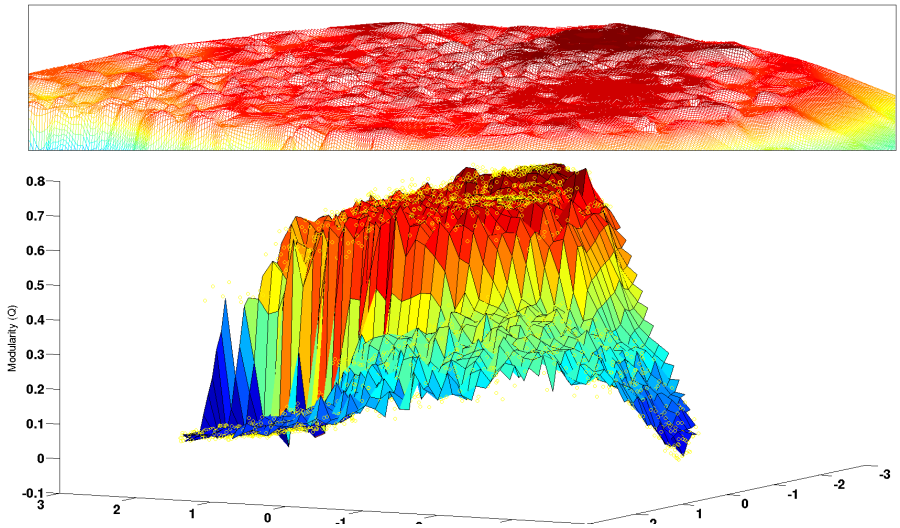


Figure 2.3: Modularity landscape of the E coli metabolic network [54] showing 3708 sampled partitions. Distance between partitions is calculated using variation of information [84] and dimensionality reduction is performed using curvilinear component analysis [85]. The inset (top) demonstrates the landscape of the high modularity region. Figure generated with the software of Good et al. [81].

In the excellent work of Good *et al.* [81] the modularity landscape of several real world graphs is explored, demonstrating the extreme difficulty of this problem. The search space for modularity optimisation is highlighted in Figure 2.3, which shows the modularity scores of 3708 partitions of an *E coli* metabolic network [54] (used in Chapters 6 and 5) with distances calculated using a measure of the mutual information between partitions [84]. The dimensionality of the search space is then reduced using curvilinear component analysis [85] to allow visualisation. Note that the x and y axes are unlabelled as curvilinear component analysis is a reduction in space that has no natural interpretation. The top inset of the figure is the search space of the partitions with high values of modularity, and lacks a clear, singular peak. In other words, there are a large number of locally optimal partitions very close to the global optima in terms of modularity score, but very distant from one another in terms of the

mutual information shared between partitions. In Chapter 3, we explore the mutual information between different module extraction approaches applied to correlation of expression networks highlighting the difficulty in algorithm selection.

This "glassy" search space creates issues for heuristic algorithms, as it is likely that they simply find one of many potentially high valued solutions that is not the global optima. Furthermore, modularity is known to have a resolution limit [86] in that small communities are hidden in the presence of large communities. Given this search space, it is extremely difficult to argue that a given global optima is the "correct" partition for empirical data. The locally optima solutions vary over such a large scale of the search space whilst still retaining modularity scores that are very close to the globally best solution. Good *et al.* also go onto speculate that this search space is not unique to modularity but, rather, potentially present in all optimisation problems of this form [81], though further analysis of this is required.

Bagrow addresses another issue with modularity, in that trees (i.e. acyclic graphs) appear to have extremely high levels of modularity [87]. One can consider that a tree has high levels of modularity because any partition of the space will likely compare favourably to the null model found in Equation 2.11. A key aspect of this result is that when assessing the significance of modular structure the density of the graph is important. High values of modularity do not indicate the presence of modular structure on their own, the reported value must be made in comparison to the null model.

This does not mean that modularity is not a well reasoned approach to detecting communities. Indeed, the original algorithm presented by Newman and Girvan [14] used modularity as a method of choosing a point to cut the dendrogram generated by hierarchically clustering nodes based on betweenness centrality.

A recent approach to community detection uses a, so called, message passing algorithm in order to explore the landscape of modularity, rather than just optimising to find a single solution [88]. The results reported within this work suggest that an approach of using multiple high value partitions will likely yield successful and meaningful results. The approach of using a consensus of good

clusterings to find a statistically valid approach has been attempted before [89].

Two modularity maximisation methods used in this thesis are the fast, greedy Louvain method [82] and simulated annealing [81].

Louvain Method

Here we discuss the method presented in [82] for the calculation of a modular community structure. In the initial phase each vertex is in an isolated community. For each node, communities are recursively merged such that they are placed in a module that provides the maximal, positive, change in the modularity score, ΔQ. Merges that involve negative ΔQ are always rejected. The initial phase is complete when merging nodes results in increase in the modularity score. This configuration can be seen as a local maxima that depends on the order in which the nodes are visited.

The second phase of this approach is to agglomerate these communities into nodes i.e. a cluster becomes a single vertex in a graph that links between clusters. The initial step of agglomerating nodes is applied recursively, until there is no gain in modularity score.

Two limitations of this method make it a poor choice for exploring the full landscape of modularity. As with all greedy methods, the inability to accept solutions with a negative ΔQ means that the algorithm is bound to converge on a local optima. Furthermore, the landscape of the partition space, with regards to modularity, has been shown to have a huge number of near optimal solutions that have little to no similarity with one another [81]. This implies that the Louvian approach, which is designed to find a community structure with as little running cost as possible, is a poor choice for a full exploration of the community space, but is still a useful estimate for Q_{max}.

Simulated annealing

Here we describe the approach to simulated annealing presented in [81], which aims to characterise the space of possible community structures, not just find a single global optima. Simulated annealing [90, 91] allows the exploration of the space of possible modular partitions by first selecting a random partition assignment and generating new partitions based on one of two move types,

transferring a node to a different, adjacent, community or splitting/merging existing communities. With probability p_m, two communities are selected and merged together. With probability p_s, a group is selected and split according to the minimum cut such that two sub groups are formed with the minimum number of edges between them.

At each time step t, the annealing schedule controls a temperature parameter T which is used to accept or deny a modification. Here a geometric schedule is used such that $T(t) = T_0 r^t$ for an initial temperature $T_0 > 0$ and $r \in [0, 1]$ is the ratio between temperatures at successive time steps. Selecting T_0 and r with appropriate values allows more exploration of the modularity space. The probability of accepting a new partition that has a negative change in modularity, ΔQ, follows the exponential decay of T, given by $e^{(-|\Delta Q/T|)}$. Thus, as T approaches 0 a local optimum will always be selected.

The real advantage of the simulated annealing approach is that a huge search space can be well sampled. As discussed above, in the case of modularity, Good *et al.* [81] found that for many real metabolic networks, there are many local optima that are extremely close to a globally optimal value of Q yet show little similarity to one another. The huge number of near optimal solutions means that it is very easy to find near optimal solutions to the maximal modularity problem, yet the actual communities detected will disagree on non-trivial topological properties. Combined with the resolution limit [86], in which partitions of large graphs with high levels of modularity ignore small scale communities, this creates issues for finding biologically relevant communities based on a single relevant community partition.

2.4.3 Information theoretic approaches

Alternative objective measures for partition quality come in the form of information theoretic approaches [92]. These methods focus on the idea of a random walker transitioning between vertices. For example, the Markov Clustering approach [93] operates by clustering the weighted transition matrix of a network. Similarly, the Walktrap algorithm [94] detects modules through the assumption that a random walker will get trapped within dense regions of a graph. More

recently, Rosvall and Bergstrom [75] liken the problem to the reduction of information found in a map, the cartographers trade off. Simply put, if a map contains too much information it becomes unreadable. A cartographer's role is to balance representing competing factors such as place names, topography and landmarks with the overall readability of their description. Conversely, ignoring too much of the structure of the underlying system will make the map too general.

Using a two stage compression Huffman coding technique, the map equation likens the best partition to the smallest possible compression of a graph. The objective of the Huffman coding is to give each node a unique identifier such that the network can be described in the minimum number of bits. A two level approach gives each node an identifier with inter and intra community keys. The objective is to find the partition of a graph such that the resulting compression is of minimum length.

More formally, given that the entropy of a codeword can be expressed as

$$H(X) = \sum_{i} p(x_i) log(p(x_i)), \tag{2.13}$$

we can define a partition of n nodes and m modules such that $L(\mathbf{M})$ minimises the description length of the network where $L(\mathbf{M})$ is given by,

$$L(\mathbf{M}) = q_{\curvearrowright} H(Q) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(P^i), \tag{2.14}$$

where $q_{\curvearrowright} = \sum_{i=1}^{m} q^i_{\curvearrowright}$ is the probability that a random walker switches between modules ($q^i_{\curvearrowright}$ is the probability that the walker exits module i), $H(Q)$ is the entropy of movements between modules, $p^i_{\circlearrowright}$ is the ratio of movements within module i plus $q^i_{\curvearrowright}$ and $H(P^i)$ is the entropy of movements within module i.

This approach can be seen as finding the minimum description length [95] of a graph, doing so will capture a coarse grained description of objects found within a graph, arguably the overall objective of any community detection algorithm. Intuitively, when one thinks of the most compressible graph, it is undeniably that of a fully connected component in which all nodes are connected. This assumption relates strongly to the idea of the assumption behind modular structure described in Chapter 4 in that a random graph

contains no modules. In this sense, it can be argued that a randomised graph, without dense substructures, is incompressible.

Extending the Infomap approach, Rosvall and Bergstrom recently developed a hierarchical approach that is based on fundamentally the same concept [96]. Instead of describing each node with two codewords, the compression takes place over multiple levels. To satisfy this condition, Equation 2.14 then changes to,

$$L(\mathbf{M}) = q_{\curvearrowright} H(Q) + \sum_{i=1}^{m} L(\mathbf{M}^i), \tag{2.15}$$

with each recursive level defined by the function,

$$L(\mathbf{M}^i) = q^i_{\circlearrowright} H(Q) + \sum_{j=1}^{m} L(\mathbf{M}^{ij}), \tag{2.16}$$

down to the bottom level of the granular description,

$$L(\mathbf{M}^{ij..k}) = p^{ij...k}_{\circlearrowright} H(P^{ij..k}), \tag{2.17}$$

Where $q^i_{\circlearrowright}$ applies to the transition probability to other levels of the hierarchy, and $p^{ij...k}_{\circlearrowright}$ describes the transition probability between the lowest level modules.

In this thesis, the concept of hierarchy is largely ignored. Despite being an interesting and important property of organisation, few methods exists to validate the simple question of whether a network is truly hierarchical, making it difficult to validate any such structure in the models presented in Chapter 4. Consequently, future sections only consider the bottom level modules detected by the different Infomap algorithms. In Chapter 3, we find that these different approaches are quantitatively extremely similar, though this result will be heavily dependent on network topology.

2.4.4 Statistically significant modules with OSLOM

The OSLOM algorithm (short for Order Statistics Local Optimization Method) [67] is an appealing method for overlapping community detection in that it is grounded in a definition of statistically significant clusters. In a similar vein to modularity maximisation [14], OSLOM attempts to uncover communities through decomposition of the graph into structures that would not occur at random. By using a definition of a significant cluster [97], OSLOM first discovers

clusters of communities, starting from a random assignment. This initial set of clusters could be detected first with other algorithms. These random clusters are then merged based on similarity, and any hierarchy within each of the clusters is discovered. A notable aspect of OSLOM is the notion of homeless vertices which have no community, this is distinct from other algorithms. For example, in the case of random graphs without community structure, OSLOM will assign a high percentage of nodes as homeless indicating the lack of a clear block structure.

A notable drawback with OSLOM, however, is that the processes relies on a random starting point and so will not generate the same cover each time the algorithm is run. Newer versions of OSLOM also use consensus clustering [89] to overcome this issue. The goal with consensus clustering is to search for a median partition from a set of alternatives. The method used in [89] works by building a consensus matrix which details the co-occurrence of vertices in clusters given a set of input partitions. This consensus matrix is then recursively clustered after ignoring co-occurrence below a given threshold until the partitions are in agreement.

2.4.5 Label Propagation

An algorithm proposed by Raghavan, Albert and Kumara [79], label propagation is an elegant solution for finding communities in large graphs. Each vertex starts with a unique label (e.g. an integer), recursively each vertex updates its label to the most popular label held amongst adjacent nodes. In the case where more than a single label is the most popular, a random label is chosen. The choice of a random label is largely irrelevant, a consensus forms as soon as one label becomes more popular. When all nodes have the label that is most popular amongst their neighbours, the algorithm stops. The limitation of this method is that the propagation of labels does not converge and so the algorithm does not terminate. As a consequence, asynchronous updating is applied; labels are based on the previous label held by their neighbours, the algorithm can then terminate when the labels held are those that are those maximally held by their neighbourhood.

In order to allow overlapping communities, Gregory [98] introduced a modified form of label propagation, COPRA. COPRA allows each vertex to contain more than a single label. As with the non-overlapping label propagation, the vertices are assigned labels which pass between neighbours. Instead of holding only a single label, however, multiple labels can spread to a given node. Formally, the degree of belonging a vertex i has to a given label c is quantified as,

$$b_t(i,c) = \frac{\sum_{j \in n_+(i)} b_{t-1}(j,c)}{n_+(i)}, \tag{2.18}$$

where $n_+(i)$ denotes the set of adjacent neighbours of i and t represents the time point. Essentially, $b_t(i,c)$ represents the labels that have spread to i after t time steps. This approach, however, still yields almost as many communities as there are nodes. Consequently, a threshold of $1/v$ determines whether or not i is a member of community c. Labels with a value of $b_t(i,c) < \frac{1}{v}$ are ignored at subsequent time steps making i a member of, at most, v communities.

COPRA is an interesting extension to label propagation that allows a vertex to be contained within multiple communities in a way that follows intuition about how messages may pass around networks. However, a clear limitation of this approach is that v is a free parameter with no indication as to what value a user should expect it to take.

Label propagation is an interesting approach to module detection when contrasted with the other methods reviewed here. The notion of a community is not based on any statistical or information theoretic assumption about what a module is or if it is detectable. Instead, the resulting community structure emerges as a product of simulation. Interestingly, however, this simulation includes a minimal amount of stochasticity, with the ideal community structure being some stable final state. However, problems arise when deciding to terminate the simulation, as node labels can continue to update indefinitely.

2.4.6 Summary of module detection methods

This section has reviewed global module detection approaches from the perspective of modularity maximisation, information theoretic, message propagation, and statistically grounded methods. These approaches are based on different

assumptions about how a community should be defined. Information theoretic approaches apply the idea of a message being trapped in subgraphs, exemplified by the minimum description length approach applied in the Infomap algorithm [75]. The statistical approaches of modularity maximisation [14] and OSLOM [67] have a similar conceptual basis, that a community should be considered an unlikely subgraph. The label propagation based algorithms [79, 98] take an approach that is distinct, in which clusters of nodes are detected through the idea of common groups forming a consensus. In Chapter 3, a selection of the module detection algorithms reviewed here are applied to co-expression networks.

The main aim of this thesis is to provide methods to analyse these algorithms in a realistic context. These assumptions give rise to the motivation behind the block structure in CiGRAM described in Chapter 4, where we consider a module to be indistinguishable from a random graph. In the following section, we move towards the core topological properties that make up complex networks, both biological and non-biological, before discussing a number of benchmark models that have been used to test the performance of algorithms.

2.5 The topology of complex networks

Central to the idea of this thesis is the notion of a randomised model, in which a set of nodes is connected according to some stochastic generative process. The first model of a random network was conceptualised simultaneously by Erdős-Rényi [99] and Gilbert [100]. This describes the class of *uniform random graphs*, as the probability of any two of the n vertices being adjacent can be described by a single parameter, p. The result of this form of random graph is that all nodes have roughly the same degree (number of edges), forming a Poisson distribution. Under this model, the probability of a vertex having a given degree k is,

$$p(k) = \binom{n}{k} p^k (1-p)^{n-k} \simeq \frac{z^k e^{-z}}{k!}, \tag{2.19}$$

where $z = p(n-1)$, the expected average degree of the network. The expression $\frac{z^k e^{-z}}{k!}$ is a Poisson probability [5] and the approximation in Equation 2.19

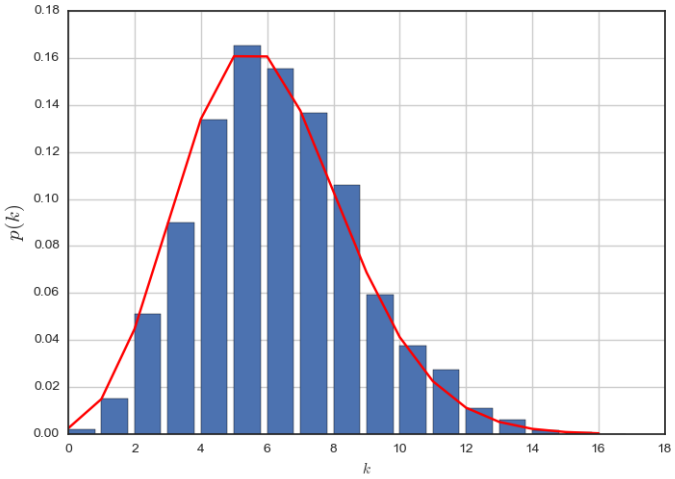


Figure 2.4: The degree distribution of an Erdős-Rényi-Gilbert network with 5000 *nodes with* $p = 0.001$. *Red indicates the estimate shown in equation 2.19*

becomes exact as $n \to \infty$, given that z is a fixed quantity. An example degree distribution for the above model is shown in Figure 2.4. Whilst conceptually simple, this model is a poor representation of the graphs found in the real world, which appear to have extremely skewed degree distributions [5].

The fixed density form of random graphs presented by Gilbert [100] are slightly different to those of Erdős-Rényi. Instead of connecting vertices with a probability, a random subset of the possible $\frac{n(n-1)}{2}$ edges are chosen. In this case, one can see p as the probability that a given edge will be sampled without replacement from the set of possible edges. Equivalently, one can see this as the removal of edges between vertices from a complete graph with probability $q = 1 - p$. This fixed density formation is the approach that is taken in Chapter 4, though extended to allow heterogeneous configurations.

The term *complex network* applies to any network with non-trivial topological properties such as heterogeneous degree distributions, latent community structure or a significantly higher than expected number of transitive relationships (triangles). Whilst the core focus of this thesis is complex biological networks, these properties are observed in fields as diverse as sociology [101], power grids [6], the internet [7], economics [102], and ecology [103].

In the remainder of this chapter we look at the types of approaches that have been used to model the structure of complex networks and observe the

interesting structural properties found within them.

2.5.1 Heterogeneous degree distributions

The uniform random graphs described above lack several key properties found in many real world networks. The degree distribution of many real world networks is often found to be extremely heterogeneous, often exhibiting a power law tail over at least two orders of magnitude [7]. In such cases, the probability density function (pdf) for the degree distribution follows the form,

$$p(k) \approx Ck^{-\gamma}, \tag{2.20}$$

where k is the degree, γ is the exponent, generally in the interval $2 \leq \gamma \leq 3$ for degree distributions, and C is the normalising constant. This approach is mainly used to express continuous distributions, which is unreliable when considering discrete data such as a degree distribution [12]. In order to express the discrete power law definition the most common approach is to use the Hurwitz zeta function given by [12],

$$\zeta(\gamma, k) = \sum_{n=0}^{\infty} (n+k)^{-\gamma}. \tag{2.21}$$

The power law distribution is undefined at $k = 0$ and requires a minimum degree k_{min} to be specified. Placing the zeta function as the normalising constant for the power law distribution, the discrete probability density function is defined as,

$$p(k) = \frac{1}{\zeta(\gamma, k_{min})} k^{-\gamma}, \tag{2.22}$$

and giving survival function, or complementary cumulative distribution

$$P(x < k) = \frac{1}{\zeta(\gamma, k_{min})} \zeta(\gamma, k). \tag{2.23}$$

Such networks are termed "scale-free" in the sense that there is no characteristic measure that can be applied to capture the scale of the network. In biological terms, this means that a relatively small number of *hub* genes account for most of the interactions within a network. This is thought to have the advantage that random, single link errors are unlikely to create issues whilst the specific targeted removal of hub nodes quickly becomes catastrophic [104].

In terms of biological networks, heterogeneous, heavy tailed degree distributions have been shown to be extremely important [22]. For example, protein-protein interaction networks are characterised by the presence of high degree "hubs" that contain an extremely large proportion of the number of connections [105]. These hub nodes are extremely important to the network structure, and their removal is shown to be catastrophic in terms of communication.

The first proposed model explaining the existence of power laws in real world networks was the Barabasi and Albert (BA) model [7]. The model has two core principles, discrete time based growth and preferential attachment. Formally, at each time step a node and m_t edges are added to the network. The probability of an existing node connecting to the new node is proportional to its existing degree,

$$p_i = \frac{k_i}{\sum_{j \in V} k_j}. \tag{2.24}$$

To highlight the importance of the combination of growth and preferential attachment, Barabasi and Albert proposed two alternative forms of model, one without growth and one without preferential attachment. Neither of these models is capable of generating a scale-free degree distribution. The role of the preferential attachment model is to propose a mechanism that explains the generative process that leads to a network forming a power law distribution. In this sense, the model puts forward a form of "rich get richer" hypothesis in which the most popular vertices have some advantage in the formation of edges. Figure 2.5 shows the contrast between a scale-free and Poisson degree distributions by comparing a BA model to an ER uniform random graph against a power law distribution with exponent $\gamma = 3$.

Whilst much of the literature is concerned with a universal property of "scale-freeness", one must be extremely careful when characterising networks with a power law, and whether the existence of one has any meaning in of itself. Stumpf and Porter succinctly point out [106] that the existence of power laws in biological datasets has been both incorrectly characterised, for example in the case of the *C elegans* metabolic networks [107], and its importance is perhaps overstated [108]. It is unlikely, therefore, that a single universal explanatory mechanism, such as preferential attachment, could be found for all

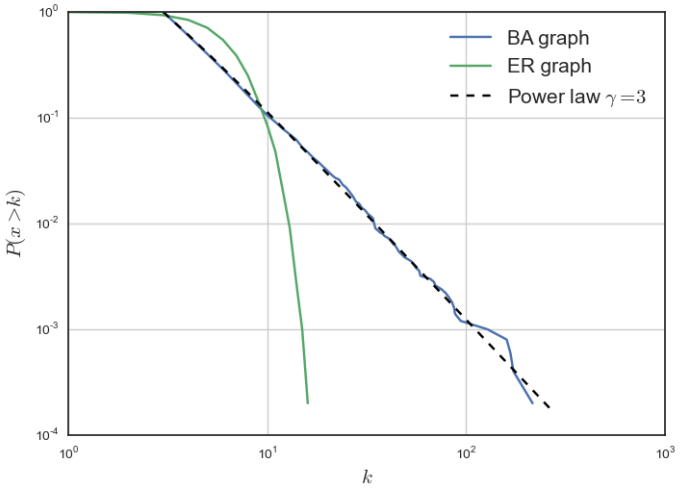


Figure 2.5: Complementary cumulative degree distributions for Barabasi-Albert (BA) and Erdős-Rényi (ER) graphs. The dashed lines indicate a discrete power law distribution with exponent $\gamma = 3$.

complex networks. What is undoubtedly important, however, is the fact that extremely heterogeneous degree distributions exist and that this has significant implications for attack tolerance [104]. In terms of biology, it means that a relatively small number of genes will have functions critical to the organisation of biological systems that may relate to core regulatory and communication mechanisms [74, 109, 110].

Highlighted in the partial gene duplication models of Chung and Lu [71], and Pastor *et al.* [70], the power laws observed in biological networks are different in form to those modelled by preferential attachment. The power law found in protein interaction networks (if one is present at all) is far steeper, with an exponent γ between 1 and 2 rather than between 2 and 3 as found in preferential attachment based models [111]. Furthermore, preferential attachment is an unsatisfactory explanatory mechanism for the evolution of biological systems where gene duplication has been proposed as the mechanism by which organisms evolve since the 1930s [112].

The partial gene duplication model of Chung and Lu [71] works as follows. Starting with a random "seed" graph, a vertex is selected at random and copied, creating a duplicate vertex. With probability p, the duplicate vertex keeps each of the original vertice's adjacent edges and with probability q an edge between

the duplicate and the original vertex forms. The model of Pastor *et al.* [70] differs in the respect that with probability q, the duplicate vertex forms an edge with any vertex in the graph node in the neighbourhood of the original.

Whatever the mechanism for the existence of heterogeneous degree distributions, they are found in many biological networks. One interesting aspect that relates to this is the idea of a "small-world" network, discussed in the following section.

2.5.2 Small worlds and transitivity

Popularised in the 1960s by Stanley Milgram's so-called six degrees of separation [113], the "small worlds" phenomenon is concerned with the notion that the shortest path between individuals within social networks is extremely small, despite the enormous size and sparsity of the networks. In uniform random graphs and scale-free networks, the mean shortest path length grows approximately logarithmically with n, e.g. $l(G) \propto ln(n)$ [114]. However, an apparent contradiction exists in many real world networks because the proportion of triangles is significantly higher than one would expect to find in a random graph of equivalent density.

An elegant model proposed by Watts and Strogatz successfully captures the high number of triangles by considering real world networks to exist somewhere in the region between order and chaos [6]. Formally, they present the model with n nodes connected to α nearest neighbours in a clockwork direction creating a perfectly regular ring lattice. For each vertex in order, the edge to its first closest neighbour is rewired to connect to another vertex with probability p, this process repeats up to the α nearest neighbours connected in the ring lattice. This process can be considered as the creation of short cuts between vertices, reducing the average shortest path of the network. When $p = 0$ the resulting graph can be considered completely ordered, but has a relatively high mean shortest path length, and when $p = 1$ the graph is equivalent to a fixed density Erdős-Rényi configuration. In the range $0 < p < 1$ the graph has higher than expected clustering but, at the same time, a relatively low mean shortest path length.

However, the major limitation of the Watts-Strogatz model is that it lacks any mechanism to generate heterogeneous degree distributions modelled by the gene duplication and preferential attachment algorithms we have seen. Whilst this does not diminish the conclusions of the model it does limit its applicability for many of the networks that are studied in this thesis, which have heterogeneous degree distributions.

2.5.3 Models with fixed degree distributions

An extremely common practice within the study of complex networks is to use the so-called configuration model, in which the degree distribution is treated not as a stochastic property configurable with a set of network parameters but, rather, a fixed quantity. This follows two such forms, the Chung Lu model [115] which can be seen as a weighted Erdős-Rényi graph, and the fixed configuration model of Molly and Reed [116] in which an exact degree distribution is constructed.

In the Chung Lu model, the probability that two nodes form an edge can be expressed as

$$p_{ij} = \frac{k_i k_j}{2m}, \tag{2.25}$$

where k_i is the expected degree of node i and $m = |E|$ the number of edges. The resulting degree distribution of a network should approximately fit a real world network. The Chung-Lu model is a good approach to generating graphs with a specific degree. It does, however, suffer from the problem that the probability, described in 2.25, allows for self-loops. This, however, is shown to be insignificant in the limit $n \to \infty$ [115].

Other approaches exist to create network structure with prescribed degree distributions. Here, the wiring process is an algorithm designed to satisfy the specified degree distribution. These approaches rely on the degree distribution being graphical, that is to say, the degree distribution must create a valid graph. A simple example of an invalid degree sequence is the set $\{2, 2\}$, as each vertex requires two edges and there are only two nodes in the graphs. When rejecting self loops, this configuration is not allowed. The method used by Blitzstein and Diaconis [117], for example, exploits the Erdős-Gallai theorem [118]. At

each stage of configuration, the algorithm can check if the current state will generate a structure that is non-graphical. This avoids the problems that occur when wiring algorithms converge to non-graphical solutions, requiring some form of backtracking which can quickly become expensive.

The most widely used algorithm for the purpose of rewiring is presented by Newman [5] and the term *configuration model* is widely used to discuss the ensemble of all possible graphs with a given degree sequence. These models may offer no explanatory mechanism for the structure observed in real world graphs, however, they offer an insight into the significance of topological properties, forming a null model. Generally we will consider the Chung-Lu approach as the most appropriate approach for this thesis as it is both efficient and the networks we work with are taken from noisy domains, making an exact degree sequence add unnecessary bias.

The use of degree specific null models allows us to check if non-trivial structures, such as a high clustering coefficient are statistically significant. The simplest approach to do this is to generate an ensemble sample of graphs generated under the Chung-Lu model, test the summary statistic in question on the generated topology and compare the distance between the empirical observation and the distribution found in the ensemble.

2.5.4 Assortative networks

Whilst the degree distribution is an important aspect of networks, it is only a single measure of potentially extremely rich and diverse topologies. In this section, we explore the property of assortative correlations within networks, that is, the propensity for nodes to connect to neighbours with similar degree [119].

Newman first proposed measuring the assortative configuration of large scale networks through use of the Pearson correlation of degree distributions [119]. We describe a network, or node within a network, to be assortative if it connects to nodes of a similar degree (e.g. high degree nodes connect, predominately, to other high degree nodes). Similarly, a network is said to be disassortative if nodes have an increased propensity to have edges with a degree different to their own (e.g. high degree nodes connect to low degree nodes).

Formally, we present the assortative degree coefficient, r, of a network with the definition given in [11],

$$r = \frac{\frac{1}{m}\sum_{j>i} k_i k_j A_{ij} - [\frac{1}{m}\sum_{j>i}\frac{1}{2}(k_i + k_j)A_{ij}]^2}{\frac{1}{m}\sum_{j>i}\frac{1}{2}(k_i^2 + k_j^2)A_{ij} - [\frac{1}{m}\sum_{j>i}\frac{1}{2}(k_i + k_j)A_{ij}]^2}, \tag{2.26}$$

where k_i is the degree of a node i, m is the number of edges in a network and A_{ij} is the binary matrix indicating the adjacency of i and j. A network is said to be assortative when $r > 0$ and disassortative when $r < 0$. There is no correlation between the degree of vertices where $r \approx 0$. In Chapter 3, the level of assortativity in co-expression networks is shown to be extremely high, indicating that this is a topological property that should be accurately modelled.

However, assortativity has received far less attention than other network properties in terms of modelling. Erdős-Rényi and Barabasi-Albert models, for example, generate graphs such that $r = 0$ [5] implying that assortativity and disassortativity are non-trivial properties that are not just influenced by degree distributions. Many of the models that generate assortative links are either based on re-wiring strategies [119] or use of p^* models that generate graphs with desired topological properties through Markov re-sampling [120–122]. However, the parameters of the model do not lend themselves to any intuition behind the graph but, instead, gives a sample of graphs that will have similar desired topological properties making it useful for statistical inference but giving little indication of how a given topological property influences network dynamics.

Despite the lack of attention in terms of modelling assortativity, some work has gone into the analysis of how assortative and disassortative structures have been shown to impact networks in interesting ways. For example, Brede and Sinha [123] showed that disassortative networks appear to be more resilient to attack (the target removal of nodes) than assortative forms. Furthermore, assortativity is known to influence so called spreading dynamics within networks [124, 125]. If hubs are connected to other hubs then it seems likely that information will pass more quickly around the network. There is certainly more interest in assortative connections, but the lack of models to effectively control the property along with other network statistics appears to hold back further analysis.

Assortativity is a correlated property that can go beyond vertex degree and can be thought of as a correlation between any similarity that the vertices may have. Papadopoulos *et al.* [126] presented an approach to modelling networks based on the idea of vertex similarity modelled in a hyperbolic geometric space. Here, the growth of the preferential attachment algorithm [7] competes with the idea of a preference vector, modelled with a point in a hyperbolic space. Several years earlier, Quayle *et al.* [127] presented a model largely ignored by the literature that connects vertices either by preferential attachment, or through the similarity of preference vectors. Whilst this is not explicitly modelling a geometric space, the assortative groups allow a clear model of community structure generating graphs with high clustering coefficients as well as small-world, scale-free topology, matching the idea of assortative grouping modelled by Papadopoulos *et al.* [126]. These approaches appear interesting, though little of their influence appears in the benchmark models for module detection algorithms reviewed in the following section.

2.5.5 Benchmarking models for module detection algorithms

Whilst there are hundreds of methods to detect modules in complex networks, there are very few methods to statistically validate and test the results of these algorithms [8]. Here we review the Lancichinetti-Fortunato-Radicchi (LFR) benchmark [9, 128] model, the current gold standard for network cluster evaluation. Alternative approaches include relatively trivial models [129] and stochastic block models [130], which are also used to detect community structure. The objective is to create a *ground-truth* set of modules that can be used to evaluate community detection algorithms. Communities in real world networks are not uniform in size [131], and, as previously discussed, degree distributions are heterogeneous; the LFR benchmark seeks to accurately models these properties.

The LFR benchmark uses a fixed degree distribution generated by selecting a power law exponent γ. The community sizes are also assigned with power law exponent κ. Each node is given a degree from this distribution and is

assigned to a given community, the fraction of edges inside its own community is determined by μ. A node is only assigned to a community if the community size exceeds its maximum degree. The generation algorithm also includes rewiring to ensure all nodes have their properly assigned degree. The evaluation of algorithms is generally tested with information theoretic measures such as normalised mutual information [132] or variation of information [84].

The adoption of the LFR benchmark is almost universal in the evaluation of modern algorithms. The assumptions that any synthetic benchmark rests upon are crucial for future research. As algorithm designers attempt to improve methods for module detection, proposals will not become widely accepted unless they perform well on widely used benchmarks. Unfortunately, it is not entirely clear that the structural communities generated by the LFR benchmark are representative of what one would expect in real world graphs [133]. The LFR benchmark also excludes topological properties beyond the degree distribution, such as assortativity, which may play a large role in the structure of communities within real graphs. With this said, the benchmark has provided a good measure for the reliability of algorithms and is rightly used to select good choices.

More recently, Seshadhri *et al.* [134] presented a model for generating a predefined block structure designed to match the clustering coefficients and degree distributions of real networks known as the Block Two-Level Erdős-Renyi (BTER) model. The BTER model works by assigning nodes to a community based on their degree and connecting the communities, internally, following a uniform random graph. The communities are then connected according to the Chung-Lu model, preserving the desired degree of the nodes. This model can accurately fit the clustering of real networks and provides an additional benchmark to the LFR model.

There are a number of limitations to the BTER model, however. The model is not capable of generating configurable levels of degree assortativity; a property found in many real world graphs. The BTER model also, makes two strong assumptions about communities that aren't necessarily well justified for all graphs. *Internally*, modules are connected as uniform random graphs, a fact that is not necessarily justifiable and is distinctly different from the LFR model [9], for example. This forces a second assumption, that the communities

of nodes are determined according to degree, rather than any other property. With this said, the internal community structure being defined as a random graph is an interesting assumption. In Chapter 4 a novel approach to generating block structure is based on the assumption that suitably heterogeneous random graphs and modules are indistinguishable.

2.6 Chapter summary

This chapter has reviewed the importance of modules in metabolic, correlation of expression and protein-protein interaction networks. The detection of these modules relates very strongly to the interdisciplinary field of complex networks. The field of community detection has an extremely wide variety of approaches to uncovering underlying modular structure, with few widely agreed upon assumptions about what an underlying module is. The chapter then reviewed models for the heterogeneous, assortative and transitive nature of real world networks, finding that the benchmarks for empirical datasets lack the ability to model all of these qualities.

2.7 Conclusions from the literature

This chapter has reviewed the clear motivation for uncovering reliable modular structure in complex biological networks. There is an apparent desire to understand how metabolic pathways function with one another, what protein complexes may exist and how genes are related under varying environmental and experimental conditions. Module detection approaches have been shown to be extremely effective in this area, offering opportunities to elucidate biological function. However, the methods to uncover the modular structure of complex networks lack any wide agreement upon what a module actually is. A clear, specific definition of a module that can be modelled and tested is required. Furthermore, the lack of agreement amongst algorithms in terms of the definitions of modular structure likely means that different algorithms are well suited to different network topologies (i.e. community density, different heavy tailed degree distributions or degree assortativity may influence algorithm

performance). Methods to evaluate this are required and current benchmark models lack the ability to accurately mimic graph structure such as degree assortativity. Moreover, little to no testing appears to have been conducted as to if assortative degree patterns impact the performance of module discovery approaches.

Chapter 3 now moves on to evaluating the performance of algorithms in the context of coexpression networks. This complements much of the literature reviewed here as methods for evaluating detected modules with experimental knowledge, phylogenetic mapping and gene ontology are explored.

Chapter 3

Modules in correlation of gene expression networks

3.1 Introduction

One important goal of plant systems biology is to elucidate the function of genes through analysis of large scale datasets [38]. This requires the development of widely available, well understood tools for both analysis and visualisation. The objective of this chapter is to evaluate existing methods in extracting meaningful information from biological networks, framing the later work of this thesis in the context of the field. With only around 40% of *Arabidopsis thaliana* genes functionally annotated based on experimental evidence [135] and even less annotation in other organisms, methods that predict the function of genes are desperately required to aid hypothesis generation for future knowledge [38].

In the context of complex networks, there are a vast array of community detection algorithms that have the potential to aid biological discovery [8]. However, many of these tools are designed for use in a general context and little work has been conducted into which algorithms perform well in gene co-expression networks. We can see the creation of a gene co-expression network as an abstraction that relates the pattern of interaction between genes. Clustering of this data allows one to identify related modules of genes which an be enriched by external sources of information such as gene ontology or pathways. Key genes within these modules can then be identified, providing the potential for

hypotheses that can be experimentally validated [44].

The detection of functional modules within biological networks is not a trivial task. MCODE clustering is a popular example widely used due to its ease of use and availability within the Cytoscape network visualisation tool [136]. More recently, a number of authors have applied algorithms from the field of community detection to biological networks. For example, recent work on the Arabidopsis protein-protein interaction network uncovered clusters with the link communities method [4]. In terms of co-expression networks the application appears to be more limited, however, some authors successfully applied modularity maximisation [137, 138] and link community detection [139] to correlation networks indicating the potential of the techniques. Note that the methods analysed here are distinct from many conventional clustering algorithms study, which generally consider some underlying distance between elements in a metric space [140]. These methods are designed to detect clusters in graphs. Furthermore, none of the approaches analysed here require the user to make judgements about the number of clusters that exist within the data.

The main goal of this chapter is to evaluate the level of agreement of community detection algorithms in the domain of plant correlation of expression networks by evaluating three *Arabidopsis thaliana* datasets and one Tomato fruit ripening dataset. Fundamentally, the objective is to answer the research question: how do different community detection algorithms compare to one another in a practical context? A secondary goal is to explore how these methods can be useful to bioscientists by providing web visualisations and generating meaningful hypotheses. The main contributions highlighted in this chapter are as follows:

- **Topological analysis of datasets.** The networks under study are compared to a number of models for complex network models to evaluate clustering coefficients, degree distributions and degree assortativity coefficients (see Chapter 2, Section 2.5).

- **Similarity of detected clusters** This chapter highlights the lack of agreement in algorithms by analysis of a mutual information measure, showing how difficult it is for researchers to select a single "best" clustering.

This analysis also compares the consistency of the clustering algorithms across a range of correlation thresholds used for constructing the networks, an important consideration given the source of data involved.

- **Use of meta-data.** The analysis then turns to the inclusion of external data in the form of gene ontology [141], phylogeny [142] and knock-out experiments [143–146] in order to explore methods for validating clusters using meta-data. This tests to see if evaluation of this meta-data can be useful as an aid in both hypothesis generation as well as evaluating the performance of algorithms.

- **Web visualisation tool.** Appendix A, to this chapter, also describes how a novel web visualisation of large scale networks was developed to allow further exploration of this dataset.

This chapter extends the analysis conducted as part of the work of Dekkers et al. [147] as well as a second publication [148] to be submitted shortly.

3.2 Datasets

In this study we investigate the result of clustering algorithms on four whole genome microarray expression datasets; FruitNet [148] a tomato fruit ripening time series network, EndoNet and RadNet [147] tissue specific datasets taken from Arabidopsis thaliana seeds during embryogenesis and SeedNet [43] which is based on a collection of different microarray datasets associated with seed germination.

At each time point or experimental condition, a microarray sample is taken, giving an expression vector for each gene. All the networks are generated using the Pearson correlation between the expression vectors for each pair of genes. The Pearson correlation coefficient (PCC) is the measure of the linear relationship between two vectors, given by,

$$PCC(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{3.1}$$

where the vectors x and y are of length n and $\bar{x}$ and $\bar{y}$ are the means of x and y, respectively. The value for a PCC between variables will be between -1 (for

direct negative linear correlations) and 1 (for direct positive linear correlations). The correlation score can be seen as a weighted graph between genes, however, the data in these experiments often lacks replication and is therefore prone to noise [44]. The networks considered here are, thus, binary interactions where correlations above a given threshold τ are considered to be edges.

A Pearson correlation matrix, alone, can show unknown relationships between pairs of genes. Of interest here, however, is the generation of a network built on the "guilt by association" principle whereby an edge, or interaction, is said to exist based on highly correlated gene scores. To this end, selecting a correlation threshold is an important issue that varies depending on the level of noise in the datasets [44].

The FruitNet co-expression network is based on time series microarray experiments taken from the wild type tomato fruit during the ripening process. This is based on 14 time points, each of which is associated with time after one of two observed physiological states. These are mature green, in which the tomato has reached its full size, and breaker, where the tomato shows visible signs of ripening (i.e. turning from green to red). Time points are indicated as such; MG1 indicates a sample taken 1 day after the tomato reaches mature green state and BR1 indicates a sample taken 1 day after the tomato has started the breaker state. The network was constructed with a correlation threshold of 0.94 selected with the spectral method described in [149].

SeedNet [43] was generated from 8 publicly available datasets [150–157] totalling 138 whole genome microarray samples, 73 of which are associated with the non-germination of genes and 65 of which are associated with germination. A correlation threshold of 0.75 was selected as the best threshold under weighted genome co-expression analysis maximising the fit to a power law distribution [158].

EndoNet and RadNet [147] are based on time series tissue samples taken from germinating *Arabidopsis thaliana* seeds in the Endosperm and Radicle. Microarrays samples were taken at 29 time points starting with a dry seeds towards the completion of seed germination. Both networks were constructed with a correlation threshold of 0.932, as selected by weighted genome co-expression analysis [158].

A major limitation for all these datasets is the lack of replication for each sample point. At each of the time points in FruitNet the value taken is the mean of 3 different replicates, whilst EndoNet and RadNet use 4 replicates for each sample. This level of replication means that there is a relatively high chance for observational error. Microarrays of this form, however, are mainly considered for hypothesis generation, which would need to be backed up with other forms of experimental analysis [44].

One aspect of the networks is that they do not form complete connected components. Whilst this is not an issue for many forms of analysis, in the case of objective functions in community detection, the lack of edges between disconnected groups can bias the procedure. As a consequence we only consider the largest connected components in this analysis. In the case of FruitNet, there are two large components, one containing 4483 genes and the other containing 3885 genes. Here we consider both, but treat each component independently in the clustering process. The different connected components are broadly associated with up or down regulation following the mature green (MG) developmental phase.

Having introduced the datasets that shall be used in this study, the next section discusses the topology of the observed co-expression networks in relation to widely used graph topology generators.

3.2.1 Topology and model fit

Whilst topological properties offer interesting insights into the structure of networks, these measures only really have meaning in the context of randomised models. For example, the clustering coefficient of the networks may appear high but this may simply be a product of the overall network density. To understand if the clustering coefficient has any impact on graph structure it must be understood in the context of appropriate null models. For the purpose of this analysis we observe the degree distributions, clustering coefficients, degree assortativity coefficients and modularity of the networks. In Table 3.1 we show the topological properties of real networks when compared with Erdős-Rényi random graphs, Chung-Lu degree fit and Barabasi-Albert preferential

Network	Model	n	m	Density	C	r	Q_{max}
EndoNet	Observed	7662	577791	0.02	0.603	0.436	0.668
	Barabasi-Albert graph	7662	569025	0.019	0.055	0.008	0.067
	Erdős-Rényi graph	7662	578660	0.02	0.02	-0.001	0.069
	Chung Lu graph	7462	577090	0.021	0.09	-0.002	0.058
RadNet	Observed	7106	586704	0.023	0.62	0.376	0.662
	Barabasi-Albert graph	7106	582909	0.023	0.063	0.007	0.062
	Erdős-Rényi graph	7106	587745	0.023	0.023	-0.0	0.068
	Chung Lu graph	6917	585786	0.024	0.104	0.0	0.053
SeedNet	Observed	8485	501522	0.014	0.502	0.177	0.561
	Barabasi-Albert graph	8485	497134	0.014	0.044	0.005	0.075
	Erdős-Rényi graph	8485	503307	0.014	0.014	-0.002	0.081
	Chung Lu graph	8099	500712	0.015	0.126	-0.001	0.057
FruitNet	Observed	8407	692416	0.02	0.476	0.501	0.575
	Barabasi-Albert graph	8407	682650	0.019	0.056	0.007	0.064
	Erdős-Rényi graph	8407	693888	0.02	0.02	-0.0	0.067
	Chung Lu graph	8108	692531	0.021	0.138	-0.003	0.048
Arabidopsis PPI	Observed	7169	17244	0.001	0.098	-0.083	0.728
	Barabasi-Albert graph	7169	14334	0.001	0.004	-0.062	0.533
	Erdős-Rényi graph	7169	17032	0.001	0.001	0.004	0.46
	Chung Lu graph	5849	17061	0.001	0.047	-0.055	0.384

Table 3.1: Observed topological properties of co-expression datasets. Density, clustering coefficient (C, see Equation 2.4), degree assortativity (r, see Equation 2.26) and maximal modularity (Q_{max}, see Equation 2.11 for empirical networks of SeedNet, RadNet, EndoNet, FruitNet and the BioGRID Arabidopsis thaliana Protein-Protein interaction network with associated models.

attachment based models. These models are described in detail in Chapter 2 Section 2.5. We contrast the correlation network's topology to that of the *Arabidopsis thaliana* Protein-Protein interaction network taken from the BioGRID database [34].

All of the graphs are characterised by mean clustering coefficients and maximal modularity scores that are significantly greater than those found in the random models of any form. Modular structures that cannot be explained without some dependency between the vertices and a high average clustering coefficient indicates a high degree of reciprocation between neighbouring edges [6]. It is worth noting that correlation networks naturally tend towards transitive behaviour. For example, if the expression patterns of genes a and b highly correlate and genes b and c highly correlate, it is highly likely that a also correlates with c [44].

Complementary cumulative degree distributions of the networks and associated models are shown in Figure 3.1. The degree distributions of the Erdős-Rényi and Barabasi-Albert graphs fail to show any similarity with the co-expression networks. This indicates that the graphs are neither Poissonian, or scale-free, lacking a clear power law fit over two or more orders of magnitude (more details provided in Section 2.5). The networks, however, are still characterised by the presence of extreme hub nodes as well as extremely low degree nodes.

The Chung-Lu models are weighted to match the degree distributions observed in the real network. Consequently, the visually close fits observed in Figure 3.1 are to be expected. A striking difference, however, is the lack of degree assortativity that is present in the networks.

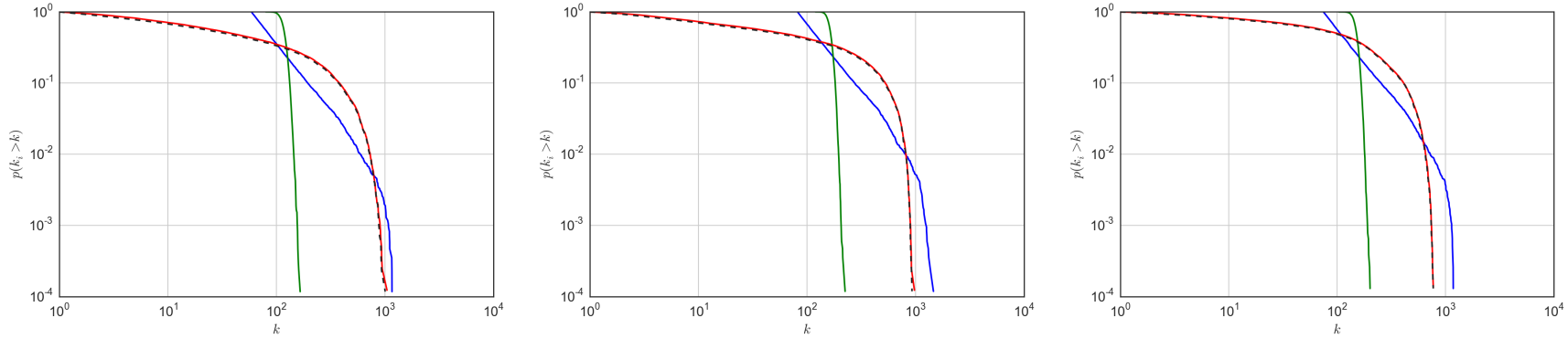
For the protein interaction network, the topology of the models is far more in agreement, proving a similar fit for the assortativity observed. Another stark difference between the datasets is the edge density, the protein interaction network is far sparser. This may be a product of an overly lenient correlation threshold used to build the networks.

Having described the datasets in terms of topological summary statistics, the following section turns to the analysis of latent community structure.

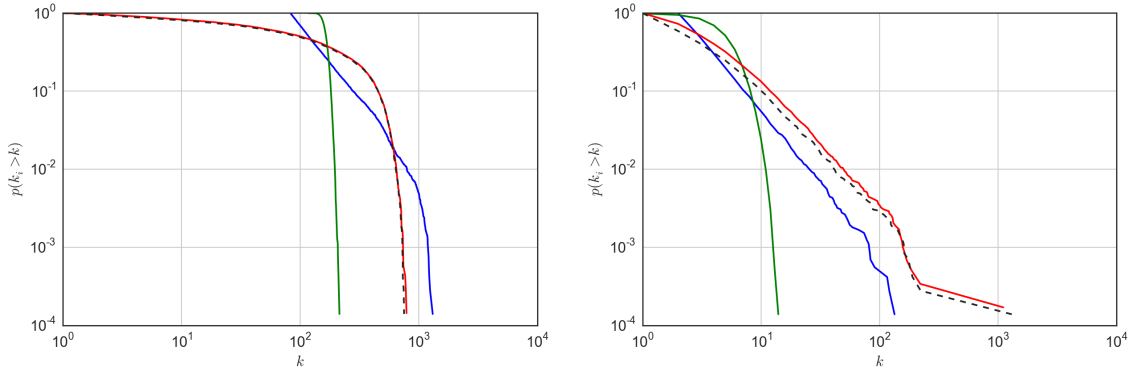
3.3 Community detection algorithms

In this section we observed the results of the clustering algorithms on the co-expression datasets. We first evaluate the lack of a strong consensus for the different clustering algorithms, making using of a normalised mutual information measure. We then make further use of normalised mutual information by testing the resilience of detected clusters to increases in the correlation threshold used to generate the network. The specific details of the algorithms used in this study are reviewed in Chapter 2 Section 2.4. Here we briefly discuss the implementations of algorithms used within this study, the algorithms are summarised in Table 3.2 which also indicates if the results contain overlapping clusters or not.

We use two forms of the infomap algorithm, the original map equation pre-



(a) SeedNet *(b) FruitNet* *(c) EndoNet*



(d) RadNet *(e) Arabidopsis thaliana PPI*

Figure 3.1: Complementary cumulative degree distributions for each co-expression network and associated models. The real graph is shown in grey dotted lines along side the Barabasi-Albert graph (blue), Erdős-Rényi graph (green), and Chung-Lu model (red).

Algorithm	Identifier	Overlapping	Implementation Notes	Citation
Infomap	info	N	-	[75]
Hierarchical Infomap	h info	N	-	[96]
COPRA	cop	Y	cop_2 to cop_7 indicates number of overlapping groups	[98]
OSLOM	OSLOM	Y	Uses consensus results from 10 runs	[67]
Louvain Modularity	louvain	N	-	[82]
Simulated Anneal Modularity	sa	N	-	[54]
Label Propagation	lp	N	-	[79]

Table 3.2: Module extraction algorithms tested in this study.

sented in [75] and the multi-level, hierarchical infomap (H. Infomap) described in [96]. For the purpose of detecting clusters the hierarchy is not relevant here and, as a consequence, we only consider the bottom level communities.

For the implementation of overlapping label propagation we use the version of COPRA described in [98]. The selection of v, the number of communities any given node can belong to, appears to be a non-trivial task; v is set at 7 different levels. The algorithm implementation is also stochastic in nature, we use the best clustering that satisfies the overlapping modularity constraint.

The OSLOM algorithm is also stochastic in nature and requires multiple runs. We use the implementation provided in [67], which includes the use of consensus clustering [89] to allow the covers with the highest level of agreement to be selected from 10 independent runs of the algorithm.

Label Propagation (lp), the Louvain algorithm and simulated annealing (SA) have no parameters that need to be user defined. The tests presented in this chapter use the implementations used to detect communities in the LFR benchmark graph models [159].

3.3.1 Comparing generated clusterings

Qualitatively it is clear that there are important differences between clusterings generated by different algorithms. To provide a quantitative measure for the differences between partitions, *normalised mutual information* (NMI) is used. The measure is presented in [160] as this variant allows us to compare covers as well as partitions. Note that the definition used here will give different results than the non-overlapping version, used for example in [9].

Our interest is to compare two different clusterings of a graph, that can either be covers or partitions, C and C', respectively. C and C' should be considered as sets containing subsets of nodes $\{1, 2..., n\} \in V$. A cluster $c \in C$ can then contain at most n nodes, and contains $|c|$ nodes. The probability of any given node belonging to c is then

$$p_c = P(X_c = 1) = \frac{|c|}{n}, \tag{3.2}$$

where X_c is the binary variable indicating such that $X_c = 1$ when a node is

present in community c and therefore

$$P(X_c = 0) = 1 - \frac{|c|}{n}. \tag{3.3}$$

We then measure the entropy for a given cluster as

$$H(X_c) = -p_c \log_2(p_c) - (1 - p_c \log_2(1 - p_c)). \tag{3.4}$$

We can then define the joint probabilities for nodes to be in the pair of clusters clusters $c \in C$ and $d \in C'$,

$$P(X_c = 1, Y_d = 1) = \frac{|c \cap d|}{n}, \tag{3.5}$$

$$P(X_c = 1, Y_d = 0) = \frac{|c| - |c \cap d|}{n}, \tag{3.6}$$

$$P(X_c = 0, Y_d = 1) = \frac{|d| - |c \cap d|}{n}, \tag{3.7}$$

$$P(X_c = 0, Y_d = 0) = \frac{n - |c \cup d|}{n}. \tag{3.8}$$

From the above probabilities, we can then calculate the joint entropy $H(X_c, Y_d)$. Our interest, though, is in the information gained about X_c given Y_d. We can express this as,

$$H(X_c|Y_d) = H(X_c, Y_d) - H(Y_d). \tag{3.9}$$

For each pair of covers, we are interested in the joint entropy between the most similar pairs of clusters. This can be expressed as,

$$H(X_c|Y) = \min_{d \in C'} H(X_c|Y_d). \tag{3.10}$$

One point to note here is that two negative clusterings will have a conditional entropy $H(X_c|Y_d) = 0$. For example, clustering the space $\{1, 2, 3\}$ into clusters $c = \{1, 2\}$ and $d = \{3\}$ has the conditional entropy of 0 despite containing none of the same vertices. As a consequence, we exclude entries from eq 3.10 if they do not also satisfy the condition,

$$h[P(1,1)] + h[P(0,0)] > h[P(1,0)] + h[P(0,1)], \tag{3.11}$$

where $h[P] = -P \log_2 P$.

In the normalised form eq 3.10 is then

$$H(X_c|Y)_{norm} = \frac{H(X_c|Y)}{H(X_c)}. \tag{3.12}$$

Giving the conditional entropy for all clusterings, $X_c \in X$, as

$$H(X|Y)_{norm} = \frac{1}{|C|}\sum_{c\in C}\frac{H(X_c|Y)}{H(X_c)}. \tag{3.13}$$

We then define the NMI between two clusterings as,

$$NMI(X;Y) = \frac{1}{2}[H(X|Y) + H(Y|X)] = \frac{H(X) + H(Y) - H(X,Y)}{\frac{1}{2}H(X) + \frac{1}{2}H(Y)}. \tag{3.14}$$

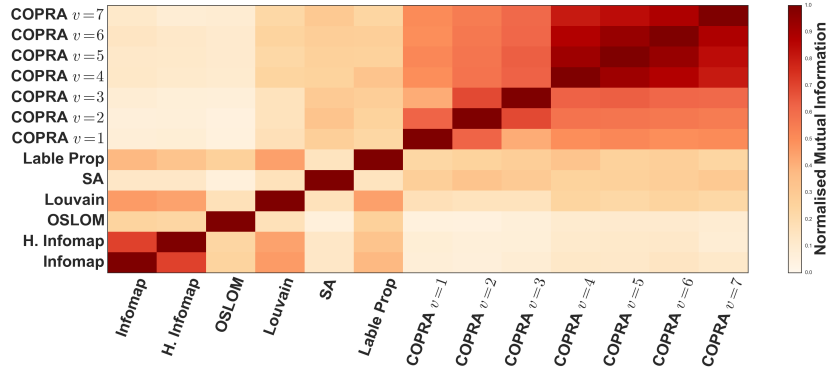
The value of I is strictly in the range $[0, 1]$ and is 1 if and only if two covers are exactly equivalent.

Figure 3.2 visually shows the NMI scores for several clustering algorithms performed on the network datasets. The algorithms based on the same method appear to have similar clusterings. COPRA, at different levels of the parameter v, appears to detect very similar communities in the cases of FruitNet and SeedNet. In the cases of RadNet and EndoNet this result appears to be less pronounced, but is widely in more agreement than other methods.

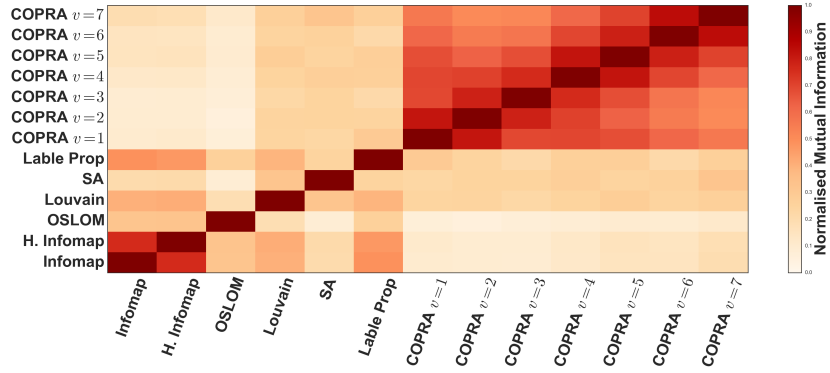
Infomap and Hierarchical infomap also appear to have very similar mutual information scores. The same, however, cannot be said about the simulated annealing and greedy agglomerative modularity maximisation methods. The low level of mutual information between the two modularity maximisers appears to conform to the results of Good et al. [81], that there are many locally optimal, high value modularity partitions that lack any real similarity.

The OSLOM approach, based on the notion of statistically significant blocks, appears to share the least consensus with other algorithms. This may be because it includes the notion of "homeless" nodes that exist between communities.

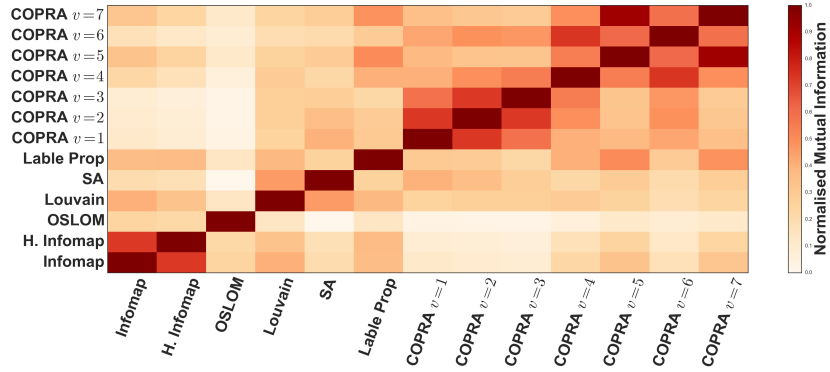
Fundamentally, different algorithms appear to show virtually no consensus between one another. The consequence is that it is difficult to justify the selection of any algorithm alone, highlighting the need of meta-data and models to assess the performance of algorithms in a domain specific context. This achieves one of the goals of the chapter; to highlight relevant limitations in module extraction algorithm selection.



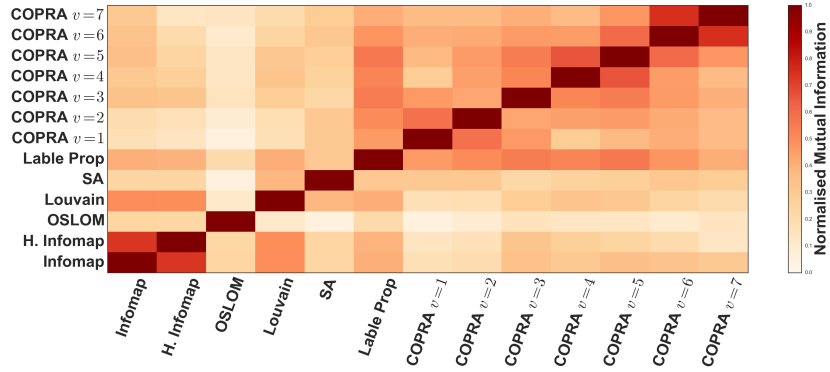
(a) SeedNet



(b) FruitNet



(c) EndoNet



(d) RadNet

Figure 3.2: Normalised mutual information between clusterings detected by different algorithms. The elements of the matrix indicate the "heat" or level of overlap between the definitions. A darker shade of red indicates that the two algorithms overlap more significantly than a lighter shade.

Robustness of coverings

The threshold selected for the co-expression networks is inherently fuzzy and prone to errors due to the limited number of samples. For this reason, we develop a method for testing the robustness of network clusterings with respect to the selected correlation threshold. This method is inspired by that of Karrer and Newman [161] in the case of testing the significance of modular structures by comparing them with random graphs. The objective here, however, is to evaluate the consistency of algorithms with respect to their initial clustering of the network. This is process is undertaken in order to evaluate any limitations in the methods when applied to these datasets. The NMI scores are measured between the partitions detected at the selected correlation threshold and an increased correlation threshold. This gives us an indication of how dependent the detected community structure is on a given correlation threshold. It is important to note that this cannot be seen as an indicator of cluster quality; an algorithm that places each node into a single cluster regardless of network topology would always score highly under this test. Instead it can be considered as a measure of consistency and resilience to spurious edges.

Results are shown for each network in Figure 3.3. RadNet appears to have the most consistent community structure, most algorithms having a high level of consistency at the first data point. Perhaps the most striking result is the change in all algorithms in SeedNet and FruitNet, to a modest increase in correlation threshold. It might be reasonable to expect that edges between clusters would be the most likely to be removed. If this were the case, however, the algorithms would have a higher level of consistency than observed here. This could be an indication that the correlation threshold is too low or there is a lack of a pronounced community structure within the network.

3.3.2 Clustering comparison summary

This section has shown the lack of agreement between the different community detection approaches under the NMI scores as well as the robustness of the respective coverings to the selected correlation threshold used for network generation. This gives an abstract overview of the problem with module

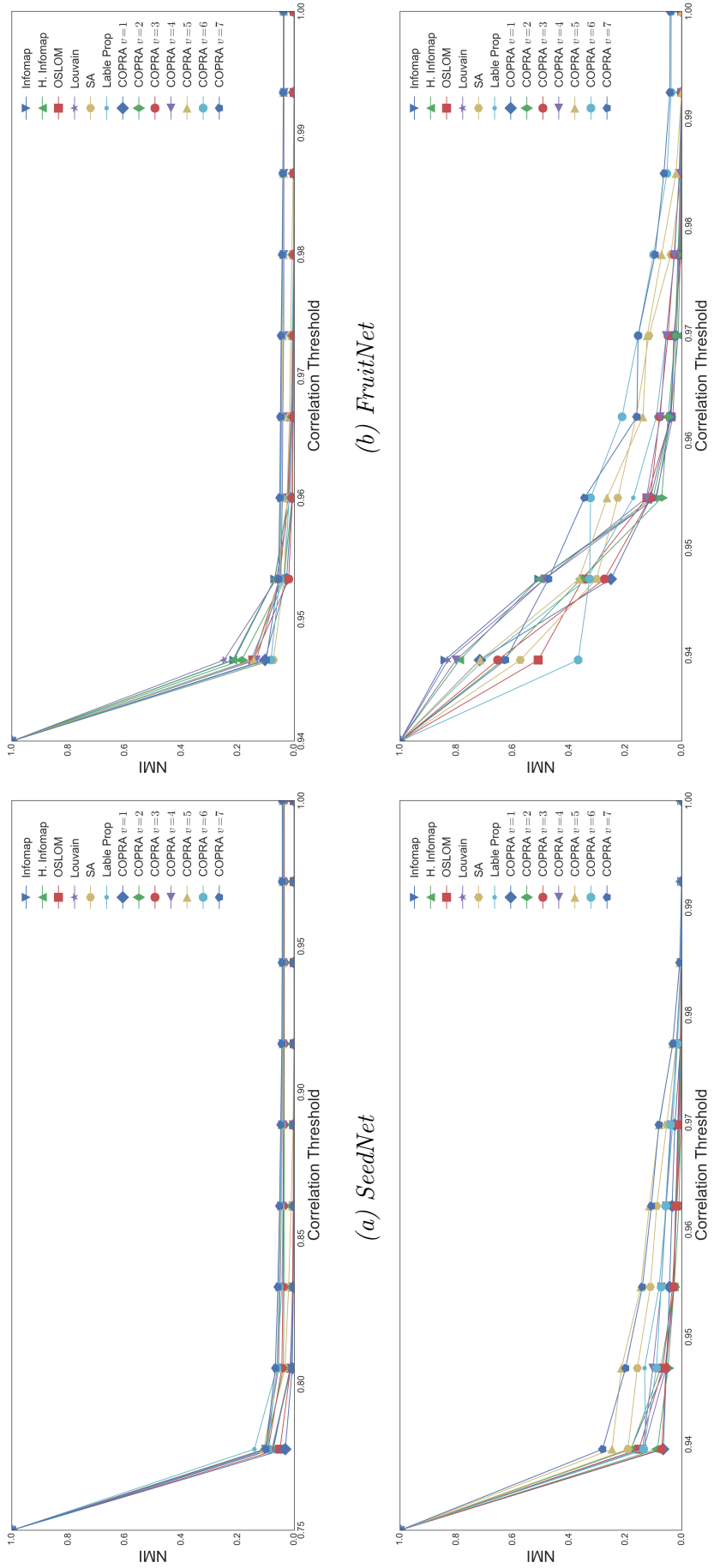


Figure 3.3: Measuring algorithmic consistency through normalised mutual information at different correlation thresholds used to generate the networks.

extraction methods in that detected modules lack a strong sense of agreement between algorithms. The evaluation of the robustness of these algorithms with respect to the correlation threshold cannot show if the algorithms perform well. However, the core aim of this work is to evaluate algorithm consistency, a feature that appears to be lacking across the range of thresholds tested.

In the following section, we provide evaluation of the detected modules through use of external data sources. These methods are popular approaches of validating clusterings within the literature, relating topological groups to known meta-data.

3.4 Enrichment of modules

On their own, clusters may provide structural information about an underlying network. Of interest is finding related, co-expressed biological modules such as functionally similar, co-regulated genes. For this, clusters must be combined with other sources of information. One of the core aims of this thesis is to evaluate the strengths and limitations of this approach in aiding algorithm selection. In order to test if the discovered clusters actually detect meaningful biological information, we combine hypothesis testing with external datasets in the form of gene ontology, a standardised vocabulary of biologically meaningful terms, [141] to validate the clusterings. The objective of this section is, then, to use a simple framework for testing to see which detected clusters are of most use to bioscientists by comparing the number of significant clusters they detect.

When a set of genes is significant, with respect to the null hypothesis, the associated term is said to be *over-represented* within the group. The most appropriate method of testing the significance of gene ontology, used in AmiGO [162], BinGO [63] and DAVID [163], is the Fisher's exact test. Fisher's exact test compares the observed number of genes associated with a given annotation. For example, a specific gene ontology term such as "DNA repair" may be extremely common within a group of nodes. To assess the significance of this result, Fisher's exact test tests the probability that one would observe this combination under the hyper-geometric distribution. More formally the probability of selecting c_s items of a given type in a cluster of size c given a

population of n genes, can be expressed as,

$$p = \frac{\binom{s}{c_s}\binom{\hat{s}}{\hat{c}_s}}{\binom{n}{c}}, \tag{3.15}$$

where $\hat{s}$ is the total number of genes not associated with the term and $\hat{c}_s = c - c_s$ is the number of genes in the cluster not associated with the term. We can say that a given gene ontology term or set of genes associated with a given pathway is significantly over expressed if it rejects the null hypothesis that the same number of genes with a given term could be found in a randomly generated subset from the population.

Testing a large number of annotations will increase the number of false positives that may occur [164,165]. In practical terms this means that a selected p-value for significance may be too lenient. The more tests that are performed the higher the probability of false positives, or so-called type-I statistical errors where a result is insignificant but still rejects the null hypothesis. The Benjamini-Hochberg corrected p-value for multiple hypothesis testing is a widely used approach to calculate the *false discovery rate* [166]. The false discovery rate refers to the fraction of false positives, that is to say the number of terms that erroneously reject the null hypothesis at a give p-value. In the Benjamini-Hochberg procedure, a q-value is set as the maximum desired false discovery rate (analogous to the p-value). The procedure works by sequentially ranking the statistical tests by their p-values (the lowest p-value being the most significant). Given the ranked values from s_i to s_n, where n is the number of tests performed, results that satisfy the condition $p > i\frac{q}{n}$ are considered significant. For example, if 100 independent hypotheses are found significant at $p < 0.05$, we may set a false discovery rate of $q = 5\%$. Under these conditions we would reject the 5 least significant results and adjust the p-value accordingly.

3.4.1 Gene ontology enrichment

Gene ontology (GO) is a controlled vocabulary used to described the role of genes within organisms. GO is really three separate ontologies: molecular functions (MF), biological processes (BP) and cellular components (CC); the structure of the organisation is hierarchical in nature forming a directed acyclic

graph. In our analysis, we consider all of the parent GO terms associated with a given term (with the exception of the three broad categories to which all GO terms belong). This means that, whilst the clusters may have very specific terms for individual genes, the categories that they belong to can also be appropriately grouped.

If the objective of clustering data is to aid understanding, smaller clusters are surely easier for comprehension. At the same time, however, these need to be relevant and related to meaningful information from external sources. Tables 3.3 to 3.6 highlight the results of the community detection algorithms in terms of several factors. The number of clusters, their mean size and variance, as well as the percentage of communities significantly enriched for least one GO term are shown.

For EndoNet, RadNet and SeedNet, the OSLOM algorithm appears to present the most useful results capturing a large number of communities that are relatively small in size with around half expressing a meaningful GO term. In contrast, the COPRA algorithm appears to generate a small number of very large clusters in the Arabidopsis datasets. Even though these clusters appear to contain meaningful Gene ontology, subgraphs this size are probably not a useful description of the data.

In the case of FruitNet, it is very important to note that the Tomato is a far less researched organism than Arabidopsis, meaning that the GO coverage is much more limited. This makes it very difficult to judge algorithms in these terms, particularly if they detect a high number of small communities. This contrasts with the larger communities detected by SA, of which 14% are enriched for at least 1 GO term. Under FruitNet, the COPRA algorithm performs very differently to the Arabidopsis datasets, detecting many more smaller communities, with relatively good rates of coverage. This is surprising considering how similar the topologies of the networks appear to be.

Of note in Tables 3.3 to 3.6 is the variance in the size of detected clusters that appear to be dominated by a small number of very large clusters, with most of the clusters being extremely small. The actual sizes of the clusters varies between the algorithms but the standard deviation in cluster size (cluster size std) appears to be large for almost every algorithm and in all the datasets.

Algorithm	clusters	Mean cluster size	cluster size std	GO	MF	BP	CC	Total Significant	p-value
Infomap	64	111.03	342.75	46.88%	29.69%	40.63%	29.69%	3738	0.01503
H. Infomap	95	74.8	201.9	47.37%	25.26%	41.05%	27.37%	3894	0.01282
OSLOM	138	60.26	46.46	89.13%	62.32%	85.51%	57.97%	5693	0.008191
Louvain	66	107.67	302.09	33.33%	16.67%	33.33%	18.18%	3556	0.01552
SA	9	789.56	588.06	66.67%	66.67%	66.67%	66.67%	3278	0.01748
Lable Prop	16	444.13	746.25	68.75%	50.0%	68.75%	50.0%	3113	0.01736
COPRA $v = 1$	36	197.39	540.78	30.56%	25.0%	30.56%	22.22%	3088	0.01717
COPRA $v = 2$	32	222.97	575.05	31.25%	25.0%	31.25%	28.13%	2943	0.01556
COPRA $v = 3$	21	360.1	719.06	52.38%	42.86%	52.38%	42.86%	3176	0.01759
COPRA $v = 4$	15	519.87	922.34	60.0%	46.67%	60.0%	46.67%	3031	0.0176
COPRA $v = 5$	7	1149.14	1105.7	85.71%	71.43%	85.71%	85.71%	3097	0.01838
COPRA $v = 6$	6	1331.17	1368.99	100.0%	83.33%	100.0%	83.33%	2706	0.01654
COPRA $v = 7$	4	1910.75	1535.16	100.0%	75.0%	100.0%	75.0%	2167	0.01575

Table 3.3: RadNet significantly over-represented gene ontology terms. Fraction of clusters with significantly expressed for one or more Gene Ontology (GO) terms with sub categories, molecular functions (MF), biological processes (BP) and cellular components (CC). Displays corrected p-values at false discovery rate set to $q = 0.05$. Null hypothesis is that the same number of gene ontology terms could be found at random, given the distribution in the population.

Algorithm	clusters	Mean cluster size	cluster size std	GO	MF	BP	CC	Total Significant	p-value
Infomap	96	79.81	244.05	46.88%	29.17%	40.63%	29.17%	4820	0.016
H. Infomap	127	60.33	186.24	44.88%	28.35%	40.16%	25.2%	5197	0.0153
OSLOM	200	45.53	35.65	84.5%	54.5%	80.5%	49.5%	7290	0.009141
Louvain	84	91.21	289.95	30.95%	17.86%	27.38%	17.86%	4438	0.01849
SA	12	638.5	716.7	50.0%	50.0%	50.0%	50.0%	3715	0.0188
Lable Prop	15	510.8	878.26	73.33%	53.33%	66.67%	46.67%	3575	0.01972
COPRA $v = 1$	40	191.55	585.3	20.0%	20.0%	20.0%	17.5%	3740	0.02095
COPRA $v = 2$	68	113.25	457.45	16.18%	13.24%	14.71%	13.24%	3845	0.02111
COPRA $v = 3$	50	161.64	552.38	14.0%	14.0%	14.0%	12.0%	3705	0.02015
COPRA $v = 4$	19	441.79	866.52	42.11%	42.11%	42.11%	42.11%	3823	0.0208
COPRA $v = 5$	7	1190.43	1284.09	85.71%	85.71%	85.71%	71.43%	3395	0.02092
COPRA $v = 6$	14	606.5	1097.99	50.0%	50.0%	50.0%	50.0%	3578	0.02158
COPRA $v = 7$	7	1240	1302.92	85.71%	85.71%	85.71%	71.43%	3552	0.02039

Table 3.4: EndoNet significantly over-represented gene ontology terms. Fraction of clusters with significantly expressed for one or more Gene Ontology (GO) terms with sub categories, molecular functions (MF), biological processes (BP) and cellular components (CC). Displays corrected p-values at false discovery rate set to $q = 0.05$. Null hypothesis is that the same number of gene ontology terms could be found at random, given the distribution in the population.

Algorithm	clusters	Mean cluster size	cluster size std	GO	MF	BP	CC	Total Significant	p-value
Infomap	191	44.42	273.36	33.51%	15.18%	28.27%	16.23%	3709	0.01366
H. Infomap	249	34.08	178.62	40.16%	19.28%	33.33%	22.89%	5318	0.01501
OSLOM	263	37.19	34.67	76.05%	51.33%	69.96%	49.81%	9658	0.01156
Louvain	178	47.67	268.19	21.35%	9.55%	17.42%	11.8%	3782	0.01694
SA	24	353.54	741.21	20.83%	20.83%	20.83%	20.83%	2835	0.01514
Lable Prop	21	404.05	1172.25	71.43%	38.1%	52.38%	33.33%	2164	0.01522
COPRA $v = 1$	91	93.24	586.28	9.89%	7.69%	9.89%	6.59%	2145	0.01471
COPRA $v = 2$	98	87.15	566.92	9.18%	7.14%	9.18%	6.12%	2241	0.01504
COPRA $v = 3$	57	152.81	760.12	12.28%	10.53%	10.53%	7.02%	2038	0.01549
COPRA $v = 4$	25	349.96	1168.16	28.0%	20.0%	28.0%	16.0%	1919	0.01419
COPRA $v = 5$	22	402	1250.64	22.73%	22.73%	22.73%	18.18%	1912	0.01561
COPRA $v = 6$	20	446.75	1317.34	25.0%	25.0%	25.0%	20.0%	1893	0.01222
COPRA $v = 7$	16	563.75	1474.9	25.0%	25.0%	25.0%	18.75%	1852	0.01228

Table 3.5: SeedNet significantly over-represented gene ontology terms. Fraction of clusters with significantly expressed for one or more Gene Ontology (GO) terms with sub categories, molecular functions (MF), biological processes (BP) and cellular components (CC). Displays corrected p-values at false discovery rate set to $q = 0.05$. Null hypothesis is that the same number of gene ontology terms could be found at random, given the distribution in the population.

Algorithm	clusters	Mean cluster size	cluster size std	GO	MF	BP	CC	Total Significant	p-value
Infomap	252	33.36	138.64	19.05%	11.11%	9.13%	7.14%	447	0.004675
H. Infomap	260	32.33	132.9	14.23%	10.0%	6.54%	6.15%	363	0.003976
OSLOM	317	30.84	33.32	22.08%	13.25%	5.99%	8.2%	236	0.001374
Louvain	377	22.3	127.94	7.96%	4.77%	3.98%	3.98%	304	0.004286
SA	62	135.6	355.31	20.97%	19.35%	14.52%	12.9%	273	0.003734
Lable Prop	66	127.38	658.14	18.18%	10.61%	7.58%	9.09%	231	0.003997
COPRA $v = 1$	652	38.68	367.72	2.91%	2.76%	1.38%	1.69%	607	0.004502
COPRA $v = 2$	960	26.41	298.62	2.29%	2.29%	0.83%	1.25%	615	0.004502
COPRA $v = 3$	785	32.56	341.61	2.68%	2.29%	1.53%	1.15%	618	0.004286
COPRA $v = 4$	418	61.13	477.27	4.07%	3.83%	1.67%	2.15%	553	0.004269
COPRA $v = 5$	305	84.34	564.28	4.92%	4.59%	3.28%	2.62%	551	0.003746
COPRA $v = 6$	174	145.9	754.89	6.9%	6.9%	3.45%	3.45%	541	0.003568
COPRA $v = 7$	131	194.11	866.53	10.69%	10.69%	5.34%	4.58%	545	0.004013

Table 3.6: FruitNet significantly over-represented gene ontology terms. Fraction of clusters with significantly expressed for one or more Gene Ontology (GO) terms with sub categories, molecular functions (MF), biological processes (BP) and cellular components (CC). Displays corrected p-values at false discovery rate set to $q = 0.05$. Null hypothesis is that the same number of gene ontology terms could be found at random, given the distribution in the population.

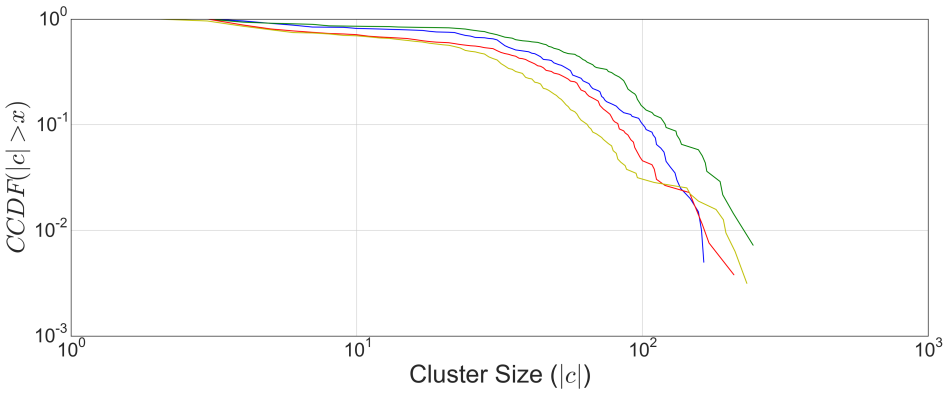


Figure 3.4: Complementary cumulative degree distributions for cluster sizes of the OSLOM algorithm on the RadNet (blue), EndoNet (green), SeedNet (red) and FruitNet (yellow) datasets.

Figure 3.4 shows this in the example of OSLOM for the different networks. When plotted on a log-log scale, the detected communities appear to be heterogeneous in nature, similar in form to the degree distributions shown in Figure 3.1.

3.4.2 Clusters and phylogeny

A recent finding in [142] showed that specific phases of embryogenesis in *Arabidopsis thaliana* correlate extremely strongly with genes that are both evolutionarily old and highly conserved in terms of genetic mutations. The implication of this result is that these phases are both extremely evolutionarily stable and, simultaneously, not robust to changes within the genome. The work of Dekkers et al. [147] shows that this also appears to be the case for germination, with specific transcriptional phases showing older genes. This section extends the work into phylogeny conducted in [147] by finding functional associations between clusters over-represented by genes within different phylogenetic categories.

Genes are divided into one of 12 *phylostrata* PS1-PS12 by using BLAST [167] to compare each gene to other organisms. PS1 contains genes that are extremely old and observed in cellular processes common to virtually all Eukaryotes and Prokaryotes. Genes in the category of PS12 are extremely young, being only observed within Arabidopsis with no homologues having matched genes. We

mirror the analysis of [147] by dividing the the *phylostrata* into three sub categories:

- PS1 and PS2 genes that arose before plant evolution.
- PS3 to PS5 genes that arose early in plant evolution.
- PS6 to PS12 genes that evolved in seed bearing plants.

The work of [147] discovered that the expression patterns of these different age groups varied throughout the time course of the experiment, confirming the previous work by Quint et al. [142]. Older genes (PS1-PS2) were found to be more strongly expressed at certain parts of the germination process, with younger genes following an inverted pattern, being expressed more before and after the germination. This indicates a phase of germination that is strongly conserved, lacking any significant expression from younger genes during crucial parts of embryo-genesis.

Because the transcriptional profiles of RadNet and EndoNet determine the edges in the network, one would expect to find clusters which mimic this pattern of containing evolutionarily old or young genes. Whilst SeedNet is not generated under the same experimental conditions as it not based on time series data, the dataset also relates to seed germination and so is included in this analysis. The objective of this work is mainly to use gene ontology to enrich clusters that are evolutionarily old or young, demonstrating the potential use of community detection approaches in biological hypothesis generation. The coverage of the phylogenetic data taken from [142] is not complete, but more than 97% of the genes in EndoNet and RadNet belong to one or more phylogenetic group, with SeedNet having over 94% coverage. The distribution of these groups is not equal, in all networks approximately 53% of the genes are associated with PS1 or PS2, 30% with PS3 - PS5 and between 12% and 15% are associated with the evolutionarily young genes in the group PS6 - PS12.

In Figure 3.5 we show the number clusters found to contain a significant number of genes by the Fisher's exact test with corrected p-values with the false discovery rate set to $q = 0.05$ as described above. Here, we count the number of clusters that are significantly represented for the phylogenetic groups

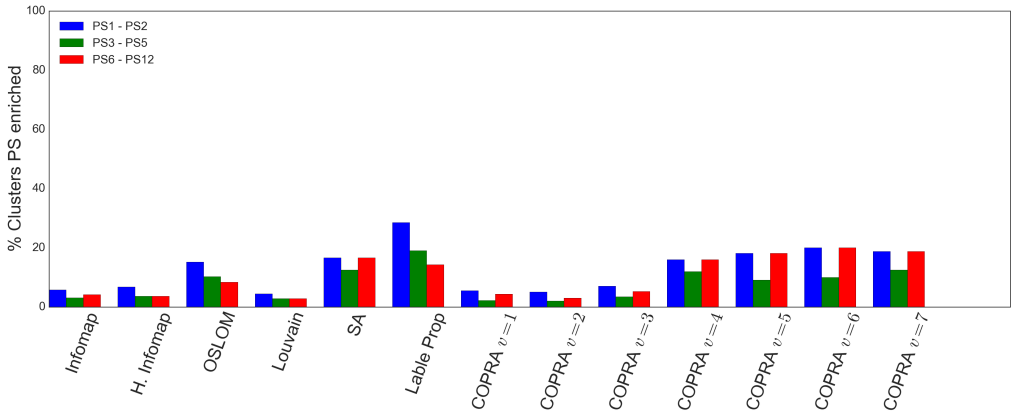
and gene ontology terms for each algorithm within the networks. The results appear to show that clusters are representative of the conserved and young evolutionary phases observed in the expression data. This is particularly the case for the COPRA algorithms which detected large, broader clusters. Unlike EndoNet and RadNet, SeedNet was not generated from time course based analysis, this could explain why SeedNet contains less pronounced significant clusters. The coverage of these clusters with GO terms also appears to be good, with nearly all the clusters that are significant for one or more PS groups also being significant for one or more GO terms.

Whilst the fraction of clusters to be enriched for a specific phylogenetic group in OSLOM was low, this can be explained by the relatively high number of communities. We observe the most significantly enriched GO terms for each GO category (BP, MF, CC) in the clusters detected by the OSLOM algorithm in Tables 3.7 - 3.9. Here we only consider clusters that are significantly over-represented by one of the phylogenetic groups; this does not appear to occur for genes in the category PS3 - PS5. EndoNet only captures a single group that is significantly over-represented by evolutionarily young genes (PS6-PS12). This group however, captures the highly relevant biological process GO term *"embryo development ending in seed dormancy"*. Whilst it is difficult to draw direct conclusions about biology, this analysis shows that the community detection approach has potential to aid hypothesis generation geared towards future work.

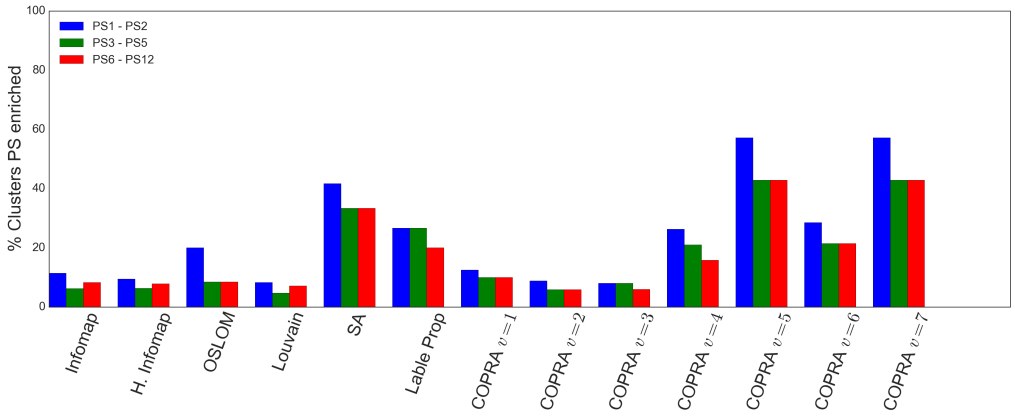
3.4.3 Knock-out experiments

As Gene Ontology appears more limited for FruitNet than the *Arabisdopsis* datasets, analysis must rely on more direct experimental data. A core objective of FruitNet is to aid the understanding of transcriptional regulation of fruit ripening. If a gene's expression profile correlates strongly with a known transcription factor it is likely that they are involved within the same process. In this section, we observe how the integration of external experiments can be useful towards finding related groups of genes.

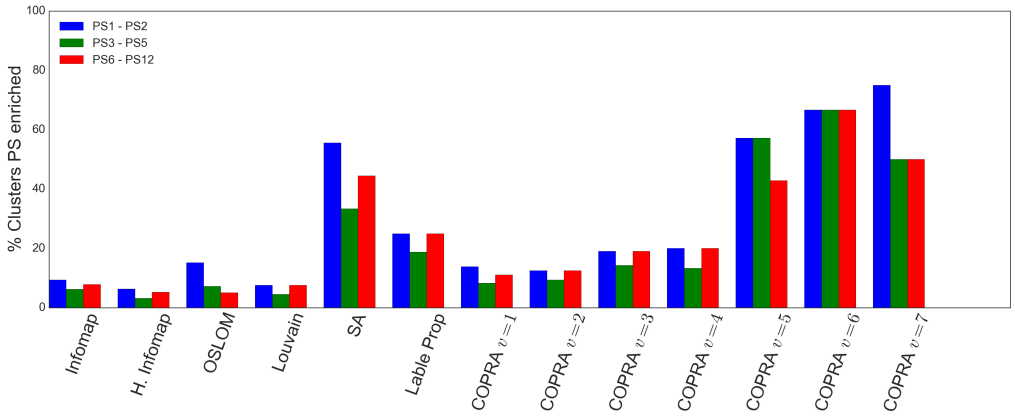
We take experimental data from 3 transcription factors known to be fun-



(a) SeedNet



(b) EndoNet



(c) RadNet

Figure 3.5: Fraction of clusters enriched for different phylogenetic groups.

Id	Cluster size	Phylostratum	BP Term	p-value	MF Term	p-Value	CC Term	p-value
134	44	PS1- PS2	mRNA processing	0.001759	nucleic acid binding	1.556e-03	nuclear body	1.752e-05
25	6	PS6 - PS12	alpha-amino acid metabolic process	0.001663	-	-	-	-
29	48	PS1- PS2	mitochondrial transport	3.509e-07	structure-specific DNA binding	5.422e-04	-	-
114	34	PS6 - PS12	RNA processing	4.852e-05	nutrient reservoir activity	1.161e-04	-	-
87	57	PS1- PS2	vesicle-mediated transport	1.859e-05	protein transporter activity	1.255e-03	bounding membrane of organelle	1.115e-06
177	102	PS1- PS2	response to metal ion	3.038e-05	threonine-type endopeptidase activity	3.799e-06	vacuolar membrane	4.981e-07
105	31	PS1- PS2	translation	2.379e-09	structural constituent of ribosome	1.097e-10	ribosome	7.938e-13
37	112	PS1- PS2	cellular process	2.687e-05	-	-	-	-
63	24	PS1- PS2	organic substance metabolic process	0.0001134	-	-	-	-
185	29	PS1- PS2	response to chemical	1.083e-05	-	-	-	-
91	59	PS1- PS2	response to UV	2.558e-05	-	-	plastid	1.744e-03
95	35	PS1- PS2	positive regulation of flavonoid biosynthetic process	3.292e-06	-	-	extracellular region	6.139e-05
152	43	PS1- PS2	ncRNA processing	0.0001045	methyltransferase activity	3.619e-04	-	-
43	65	PS1- PS2	nitrogen compound transport	4.737e-06	-	-	plasma membrane	3.585e-04
147	31	PS1- PS2	sterol biosynthetic process	2.651e-08	xyloglucan:xyloglucosyl transferase activity	4.697e-05	extracellular region	2.382e-06
148	25	PS1- PS2	macromolecule methylation	2.597e-07	ligase activity	4.032e-04	mitochondrion	1.349e-05
76	17	PS1- PS2	response to nitrogen compound	3.44e-09	carbohydrate derivative binding	6.985e-04	plasma membrane	1.702e-03
72	18	PS3 - PS5	response to chitin	2.675e-07	transferase activity	6.954e-03	plant-type cell wall	2.337e-03

Table 3.7: Most significant gene ontology terms in EndoNet for communities detected by the OSLOM algorithm significant for a phylogenetic group. Null hypothesis is that the same number of genes in a given phylogenetic category would be found at random given the population wide distribution.

Id	Cluster size	Phylostratum	BP Term	p-value	MF Term	p-Value	CC Term	p-value
8	25	PS1- PS2	response to chitin	5.066e-10	-	-	-	-
124	28	PS1- PS2	nucleocytoplasmic transport	1.223e-05	-	-	-	-
53	95	PS1- PS2	pyrimidine-containing compound biosynthetic process	2.378e-08	RNA binding	4.874e-05	non-membrane-bounded organelle	3.929e-07
86	96	PS1- PS2	translation	2.567e-19	structural constituent of ribosome	2.043e-19	ribosome	1.116e-21
38	76	PS1- PS2	nucleobase-containing compound metabolic process	7.406e-09	helicase activity	4.231e-04	membrane	7.324e-05
61	29	PS6 - PS12	embryo development ending in seed dormancy	2.893e-06	binding	4.263e-03	-	-
40	28	PS1- PS2	cellular cation homeostasis	5.372e-08	iron ion binding	6.490e-03	extracellular region	5.765e-05

Table 3.8: Most significant gene ontology terms in RadNet for communities detected by the OSLOM algorithm significant for a phylogenetic group. Null hypothesis is that the same number of genes in a given phylogenetic category would be found at random given the population wide distribution.

Id	Cluster size	Phylostratum	BP Term	p-value	MF Term	p-Value	CC Term	p-value
132	41	PS1- PS2	nucleic acid metabolic process	4.581e-06	binding	6.647e-07	nucleolus	5.691e-04
134	31	PS1- PS2	nucleocytoplasmic transport	1.105e-09	nucleic acid binding	8.871e-07	-	-
161	82	PS6 - PS12	organic cyclic compound metabolic process	1.531e-05	ADP binding	2.237e-05	plastid	6.692e-04
167	36	PS3 - PS5	DNA-templated transcription, elongation	1.006e-56	structural constituent of ribosome	3.560e-08	chloroplast	3.834e-21
80	45	PS3 - PS5	photosynthesis	3.576e-32	chlorophyll binding	6.219e-23	chloroplast thylakoid membrane	1.115e-34
81	26	PS3 - PS5	response to nitrate	5.202e-10	catalytic activity	2.951e-04	extracellular region	2.556e-08
254	32	PS1- PS2	protein import into nucleus	2.084e-12	nucleic acid binding	2.201e-04	nucleolus	1.391e-04
252	59	PS1- PS2	oxidation-reduction process	1.503e-07	coenzyme binding	1.875e-10	nucleus	1.375e-03
66	42	PS1- PS2	respiratory burst involved in defense response	3.199e-31	sequence-specific DNA binding transcription factor activity	2.800e-05	-	-
67	10	PS1- PS2	-	-	-	-	extracellular region	6.464e-04
174	40	PS1- PS2	protein targeting to mitochondrion	2.387e-12	-	-	nucleolus	3.382e-03
173	27	PS3 - PS5	protein N-linked glycosylation	1.090e-04	protein serine/threonine kinase activity	1.157e-04	-	-
196	63	PS1- PS2	regulation of unidimensional cell growth	1.660e-03	serine-type endopeptidase inhibitor activity	7.680e-06	-	-
229	64	PS6 - PS12	phosphate-containing compound metabolic process	5.581e-07	purine ribonucleoside triphosphate binding	3.377e-04	plastid stroma	4.215e-08
90	23	PS3 - PS5	fatty acid beta-oxidation	3.299e-04	-	-	-	-
117	66	PS3 - PS5	regulation of meristem growth	1.568e-09	catalytic activity	5.046e-05	membrane	1.313e-05
155	13	PS1- PS2	RNA methylation	9.657e-05	structural constituent of ribosome	9.735e-04	non-membrane-bounded organelle	4.481e-07
207	22	PS1- PS2	cuticle development	1.022e-04	carboxylic ester hydrolase activity	1.688e-03	extracellular region	2.957e-05
208	146	PS6 - PS12	protein glycosylation	4.435e-08	protein kinase regulator activity	2.843e-03	nucleus	4.738e-05
76	46	PS1- PS2	cellular metabolic process	9.504e-06	transporter activity	1.404e-03	organelle	1.319e-05
70	32	PS3 - PS5	DNA-templated transcription, elongation	2.123e-29	NADH dehydrogenase (ubiquinone) activity	1.253e-10	macromolecular complex	2.691e-10
261	93	PS6 - PS12	glucose catabolic process	9.176e-09	binding	2.104e-05	membrane coat	5.117e-06

Table 3.9: Most significant gene ontology terms in SeedNet for communities detected by the OSLOM algorithm significant for a phylogenetic group. Null hypothesis is that the same number of genes in a given phylogenetic category would be found at random given the population wide distribution.

damental to the ripening of Tomato fruit; RIN [143, 144], TDR4 [145] and APA2a [146]. We use two independent datasets for the analysis of RIN by Zhong et al. [143] and Fujisawa et al. [144]. In these datasets, a transgenic plant with the transcription factor removed is compared to a wild type plant in a so-called *knock-out* experiment. The genes of interest are those for which the expression levels are significantly impacted in the transgenic plant not containing the transcription factors.

We note that there is no reason for all the genes to be contained within the same cluster as the external experimental results do not depend on correlated expression profiles. Instead the objective is to evaluate how clustering algorithms may be of use, providing hypotheses about related genes that may not be included in the initial experiments. In Table 3.10, we show how well each gene set is represented by the algorithms. We record the number of clusters associated with the genes of interest and note the percentage of these which are significant under the Fisher's exact test with corrected p-values at the false discovery rate $q = 0.05$. All the algorithms appear to detect groupings of significant clusters, but it is important to note that, with this type of query, smaller clusters are vastly more useful.

3.4.4 Enrichment summary

This chapter has evaluated existing methods for validating clusterings detected by different module extraction approaches, a key objective of the thesis. Many of these methods show that the detected groups appear to detect meaningful clusters that relate to biological meta-data. In a comparative sense, however, these results do very little to aid method selection. Indeed, it appears that, just with the mutual information scores, the algorithms appear to lack any semblance of agreement. If the meta-data are to be useful, they must come in the form of *ground-truth* sets that can be used to test the accuracy of classifications made. Current standards of collecting data may aid hypothesis generation but offer little help when validating the algorithms designed to detect modular structure.

Algorithm	TDR4 significant	RIN (exp 1) significant	RIN (exp 2) significant	AP2a significant	p-value
Infomap	4	4	6	8	0.01035
H. Infomap	5	2	5	9	0.0176
OSLOM	10	12	14	12	0.01459
Louvain	2	3	4	5	0.01357
SA	2	4	4	3	0.02149
Lable Prop	1	2	3	3	0.01374
COPRA $v = 1$	0	6	8	6	0.01374
COPRA $v = 2$	0	6	7	6	0.02029
COPRA $v = 3$	0	6	9	6	0.01612
COPRA $v = 4$	0	6	6	6	1.701e-09
COPRA $v = 5$	0	6	6	6	1.231e-08
COPRA $v = 6$	3	6	6	6	0.03867
COPRA $v = 7$	3	6	7	6	0.03837

Table 3.10: FruitNet significant clusters found with each algorithm for known co-regulated gene sets. The number of significant clusters with associated genes is indicated with the highest p-value indicated on the right. The null hypothesis is that, given the distribution of genes related to knock-out targets in the population, the fraction of genes observed could by randomly selecting a group of equivalent size.

3.5 Chapter Summary and Discussion

The scale and size of whole genome expression data makes it difficult to understand the mechanisms that determine the distributions of data. This means that large scale statistical and machine learning techniques must be applied in order to understand systemic function. This chapter has presented a number of correlation of expression networks from plant datasets that are similar in topology and applied a variety of clustering algorithms to them. The overarching research question of this chapter was to understand how different community detection algorithms compare to one another in an applied situation. It was found that these clustering algorithms appear to lack any distinct similarity in the clusters they detected. This makes assessment of which algorithm to select a difficult choice.

Whilst the lack of agreement between algorithms may be a problem, the significant clusters that relate to known functions offer insight into biology. By combining modules found by community detection algorithms, this work shows the potential module extraction approaches have in aiding hypothesis generation as well as aiding the validation of the network structure. Indeed, the findings presented here concerning the resilience of the detected clusters to an increase in correlation thresholds may be of use in evaluating a statistically meaningful expression threshold. The idea of modular structure determining thresholds has been attempted in [149], however, this work uses conventional spectral clustering which assumes that the underlying clusters are roughly uniform in size [17, 18]. This is a property that does not appear to be matched in the results of community detection methods applied here.

The caveats associated with correlation of expression data in general, however, present significant future challenges. The notion of an edge is only based on so-called "guilt by association", (in itself a logical fallacy) that may not be as well suited as methods such as [168] that use machine learning techniques to discover novel mechanisms [169]. Furthermore, the datasets used here are based on thresholds that are inherently prone to error due to sampling bias from microarray experiments as well as a lack of replication. Future expression experiments, however, are more likely going to be based on RNA-seq data,

which overcomes many of the limitations of Microarrays such as the need to design a probe to detect every potential gene transcript [38]. This, however, will also increase the scale and complexity of the networks constructed.

Even if the methods used to construct the networks were to be modified, problems with the clustering and data analysis step need to be solved. A fundamental issue briefly touched upon in this chapter is the lack of appropriate null models for the networks in question. Whilst degree fit based approaches such as the Chung-Lu model [170] show some promise, they appear to lack other salient features found in the network such as assortativity. The current gold standard in community detection algorithms is the LFR method [9, 128], which is based on an exact degree fit whilst ignoring other topological properties. It may be possible to modify the construction algorithm to match empirical datasets, but further work is required to define and test the fundamental assumptions about what communities or clusters in networks actually are [8].

It is worth noting that the LFR benchmarks [9,128] do not include provisions for generating graphs with assortative community structure. The high level of assortativity within the correlation of expression can be explained by the fact that the use of a correlation between expression vectors gives co-expression networks a geometric nature; the Pearson correlation coefficient can be interpreted as the cosine of the angle between two vectors [171]. One can then see the selection of an edge threshold for generating a network as a distance threshold. An appropriate model may then be, a random geometric graph [172] where edges are then drawn between nodes if they are within a specified distance from one another. Chapter 4 presents a model that aims to generate the assortative, heterogeneous and highly modular topology observed in this chapter, fulfilling the goal of a ground-truth benchmark.

Chapter 4

Circular Gaussian random graph models

4.1 Introduction

In this chapter, we introduce and develop the concept of the **Ci**rcular **G**aussian **R**andom gr**A**ph **M**odel (CiGRAM). The objective of this approach is to generate synthetic networks with a ground-truth modular structure and the realistic topological properties including heterogeneous degree distributions and assortative connections found in empirical biological datasets explored in Chapter 2 and Chapter 3. The core aim of this thesis is to provide a mechanism for evaluating module extraction algorithms in the context of domain specific models. Specifically, this chapter aims to answer two core research questions:

- Alongside heterogeneous degree distributions, how can assortative structure be modelled?
- How can a module be formally defined?

CiGRAM is a novel, geometric approach to modelling the probability space that determines how edges are drawn between nodes. By using the geometry of a unit circle, giving all nodes a position about its circumference and associating certain positions with a higher propensity to form edges, it is possible to generate graphs with extremely heterogeneous degree distributions. This approach has some similarity to other approaches in modelling complex networks in that

geometry is used, but is very different in formation. In a very different geometric approach, Papadopoulos et al. [126] use distances between hyperbolic positions to determine the weights between edges. Similarly, [173] uses circular geometry to model metabolic networks. Both of these modelling approaches attempt to uncover a hidden latent geometry in networks. In this chapter no such claim is made. Furthermore, [126] and [173] use growth and attempt to mimic preferential attachment. In CiGRAM, distances and positions in geometric space are used as a convenient mechanism for generating networks and no statements or judgements are made about how this space relates to real world empirical data. Validation, in terms of the topological fit to real world networks, comes in Chapter 5.

This chapter first introduces the concept of CiGRAM for graphs without community structure, showing that degree heterogeneity can be modelled through the use of two wrapped Gaussian probability density functions, with means at diametrically opposing points on the unit circle. This is shown to be a natural extension of a fixed density uniform Erdős-Renyi-Gilbert random graph [99, 100], by modelling degree heterogeneity through geometric positions in space. The use of an underlying geometric space then allows one to include positive and negative degree-degree correlations by use of distances. Results obtained using CiGRAM indicate that positive assortativity becomes extremely difficult to model in dense configurations, indicating that it may be a property only relevant to sparse networks.

The chapter then moves on to generating overlapping, heterogeneous community structure based on the simple assumption that null random graphs are indistinguishable from communities. The approach to modelling communities shows that high clustering coefficients are a natural product of networks with modular structure. In a similar vein to earlier results, we show that positive assortativity is difficult to model in highly dense community structure, requiring communities that are internally sparsely connected, or have a high level of mixing between communities.

A web visualisation of The CiGRAM algorithm is available at `http://cigram.ico2s.org` and allows the reader to configure networks according to the process outlined in the following section.

4.2 Single community model

The following section presents the reasoning and justification behind the basis of CiGRAM, the definition of a graph without community structure. Section 4.2.1 discusses the wrapped normal distributions used to generate the heterogeneous weights that allow the model to generate heavy tailed degree distributions. Section 4.2.2 then provides the specific details of the model construction, with the aim of being sufficient to allow re-implementation. Section 4.2.3 then describes the relationship with uniform random graphs, highlighting that the weights in CiGRAM extend naturally from this definition. The details of Section 4.2.4 then show how the use of geometric positions can be parametrised to allow degree assortativity. Results of the single community version of CiGRAM are then presented in 4.2.5, highlighting the impact the parameters have upon generated topology.

4.2.1 Wrapped Gaussian distributions

At the heart of CiGRAM is the usage of wrapped Gaussian distributions to create the heterogeneous probabilities that determine the edge connectivity of the algorithm. In principle, any geometric distribution could be used in CiGRAM, the wrapped Gaussian distributions are used simply as a product of convenience. Indeed, the essential model components to CiGRAM only require positions in space associated with scoring and distance functions. Found in directional statistics [174], wrapped Gaussian distributions are standard normal probability distributions applied to the geometry of a circle via a process of wrapping the line around a circle. The wrapped Gaussian probability density function is defined for any position on the unit circle, θ as,

$$g(\theta;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} \exp\left[\frac{-(\theta-\mu+2\pi k)^2}{2\sigma^2}\right], \tag{4.1}$$

where μ is the expected value and σ is the standard deviation of the underlying Gaussian distribution.

We then defined two wrapped Gaussians with central points at opposing poles of the unit circle. Formally, we define the position probability density function $f = g(\theta;\mu = 0,\sigma)$ and a scoring function $s = g(\theta;\mu = \pi,\sigma)$. We

show these distributions in Figure 4.1. For convenience, the parameters of CiGRAM are denoted as σ_f and σ_s, for the position, f, and scoring, s, functions respectively.

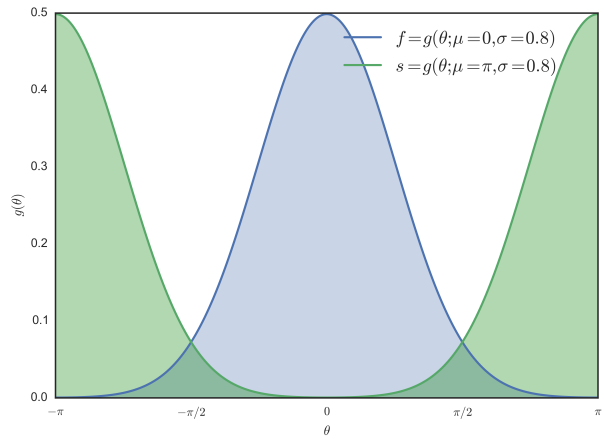


Figure 4.1: Wrapped Gaussian distributions. The probability of position (f) and scoring (s) wrapped Gaussians are centred at opposing poles of the unit circle, respectively.

Each item i in a population is given a position $\theta_i \in [0, 2\pi]$ which is a random variate drawn from a distribution with the probability density function f. The score for θ_i is defined as $s(\theta_i) = \alpha_i$. Thus, under this definition the most likely position for an item to fall $\theta_i = 0$, has the lowest possible α_i.

The Lorenz curve measures the inequality in a distribution. In the context of networks, the Lorenz curve can be used to visualise the heterogeneity of degree distributions. The Lorenz curve is generally displayed in terms of percentages (e.g. the lowest ranked 25% of the population have 10% of the total value). Given a cumulative distribution function (CDF) $F(x)$, the Lorenz curve is defined as

$$L(x) = \frac{\int_0^x F(x)dx}{\int_0^1 F(x)dx}, \tag{4.2}$$

where x indicates the fraction of items such that $x \in [0, 1]$.

Figure 4.2 shows the impact of σ_s on the Lorenz curves of α_i distribution. Where the network has perfect degree equality (i.e. a uniform distribution), the Lorenz curve is a straight line. The level of curvature can then be seen as the level of deviation from uniform equality. As $\sigma_s \to \infty$, α becomes uniform, this results in probabilities identical in form to the probability of the Erdős-

Rényi-Gilbert uniform random graphs. Further exploration of this is covered in Section 4.2.3.

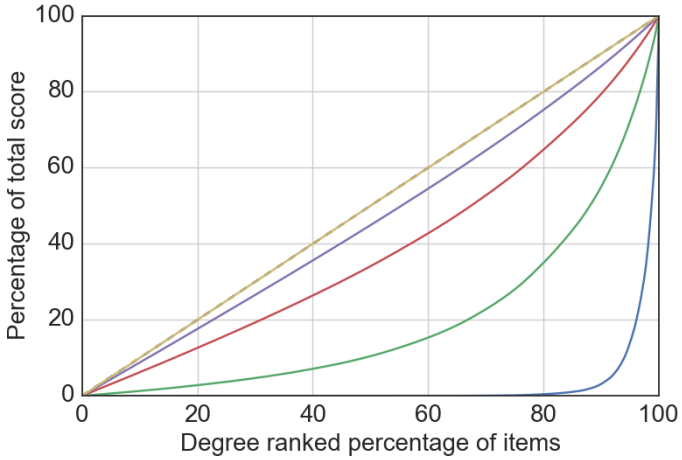


Figure 4.2: α distribution depending on σ_s. With fixed $\sigma_f = 1.0$, σ_s varies between 0.5 *(blue),* 1.0 *(green),* 1.5 *(red),* 2.0 *(purple) and* 100 *(yellow). The black dotted line relates to perfect equality.*

4.2.2 Model construction

This section defines the basic model for CiGRAM. CiGRAM uses a fixed edge density for generating a given graph. Let n denote the number of nodes and $m = |E|$ the desired number of edges in the graph. The overall objective is to select m edges from the set of $n(n-1)/2$ possible edges, given a set of weights generated using the wrapped Gaussian functions. By convention, in much of this thesis we use the measure of density, d, (see Equation 2.1) to describe the number of edges in each graph.

The reader should refer to the pseudo-code of the procedure for generating graphs in Algorithm 2. The position variables and scoring function are parameters of the algorithm. Each node is assigned a position about the unit circle, sampled from the wrapped Gaussian distribution,

$$\theta_i = g(\theta; \mu = 0, \sigma_f), \tag{4.3}$$

The weight for each node is then defined as

$$\alpha_i = g(\theta; \mu = \pi, \sigma_s). \tag{4.4}$$

In order to compute the probability of selecting a given vertex the normalised form is used,

$$\beta_i = \frac{\alpha_i}{\sum_{i \in V} \alpha_i}. \tag{4.5}$$

The weight for selecting an edge can then be defined as,

$$W_{ij} = \begin{cases} 0 & \text{if } i = j \text{ or } A_{ij} = 1, \\ \beta_i + \beta_j & \text{otherwise.} \end{cases} \tag{4.6}$$

where A_{ij} is the binary variable indicating whether or not i and j are adjacent. A naive approach would be to use weighted sampling without replacement across the normalised weights W_{ij}. A key limitation, though, is that sampling from even a modestly large set of edges quickly becomes intractable. Conventional sampling without replacement is extremely slow when probabilities are unequal and given the size of graphs that we aim to generate. In order to complete the sampling without replacement procedure we use the sampling technique of Efraimidis and Spirakis [175]. This procedure is extremely efficient as it uses exponential jumps and a reservoir to minimise the number of random variates that need to be generated.

In this process, each node is assigned a key $x_i = u^{1/w_i}$ where u is a random variate in $[0, 1]$ and w_i is the element's weight. The reservoir R is then filled with the first s elements in the population, where s is the desired number of samples. The process iterates through the list of variates, treating the lowest key, $\min_i x_i$, as a threshold for entry into the reservoir. Where the lowest key is exceeded, this element is replaced and the process continues until the population is exhausted. To further improve performance, exponential jumps are used to reduce the number of random variates to be generated [175]. In this case a random variable X_w is defined as follows,

$$X_w = \frac{\log(U)}{\log(T_w)}, \tag{4.7}$$

where U is a random number selected uniformly in the range $[0, 1]$ and T_w is the threshold of the lowest key in the reservoir. Instead of generating a key for each variable, the elements of the population are sorted in ascending order. If the sum of weights of preceding elements is higher than X_w then the lowest element in the reservoir is replaced and the new key is set based on the threshold of a

new key in the range $[T_w^{w_i}, 1]$. This process results in a reduction from $O(n)$ random variates to $O(s \log(n/s))$ where n is the population size and s is the desired number of samples [175].

A formal definition of the algorithm for the weighted sampling without replacement procedure (SampleWRS) with exponential jumps is outlined in Algorithm 1.

Algorithm 1 Weighted reservoir sampling without replacement [175]

1: **procedure** SAMPLEWRS(Population V, Sample size s, Weights w)
2: Sort population V by weights w.
3: Initialise R as the first s items in V
4: **for** v_i in R **do**
5: $u_1 = \text{random}(0,1)$
6: Calculate key $x_i = u_1^{1/w_i}$
7: Set threshold T_w as minimum key in R
8: Set $u_1 = \text{random}(0,1)$
9: Set $X_w = \log(u_1)/log(T_w)$
10: Set $W_s = \sum_{i \in R} w_i$
11: **for** population v_c not in R **do**
12: $W_s = W_s + v_c$
13: **if** $W_s > X_w$ **then**
14: Set new $X_w = \log(u_1)/log(T_w)$
15: Replace item in R with min key with v_i
16: $u_2 = \text{random}(T_w^{w_i}, 1)$
17: Set key of v_i as $x_i = u_2^{1/w_i}$
18: Set threshold T_w as minimum key in R
19: **return** Reservoir R

As the rate of growth in the possible edge set is almost order n^2, despite its efficiency, Algorithm 1 becomes intractable very quickly. As a consequence CiGRAM uses a two step selection procedure, highlighted in Algorithm 3 lines 5 to 20. Each edge is broken into two "stubs" (the individual vertices of the edge pair), the first is sampled with replacement and the second is sampled without replacement. m vertices are first sampled with replacement from the vertex set V with probability β_i. This gives each node m_i fraction of edge stubs to be completed in a secondary selection procedure. For the secondary

selection procedure we define S the matrix of weights such that the element,

$$S_{ij} = \begin{cases} 0 & \text{if } i = j \text{ or } A_{ij} = 1, \\ \hat{\beta}(j|i) & \text{otherwise.} \end{cases} \tag{4.8}$$

where the secondary selection weight for each node is given by,

$$\hat{\beta}(j|i) = \frac{\alpha_j}{\sum_{u \in \tau(i)} \alpha_u}, \tag{4.9}$$

where $\tau(i)$ is the set of nodes not adjacent to i. As the matrix S needs to be updated after every sampling without replacement procedure, it is only necessary to store $n-1$ weights at a time. This offers a considerable space performance increase, given that the matrix is neither sparse nor symmetric and contains n^2 elements.

This procedure is sufficient to generate many graphs. However, in the case of dense graphs or extremely heterogeneous configurations the maximal degree of each node can be exceeded. In this case, the procedure sets the value of $\beta_i = 0$ for all nodes with $n-1$ edges and repeats the primary and secondary selection procedures until m edges have been selected.

In certain circumstances, it is desirable for the nodes within the graph to have a minimum degree. In the above condition, disconnected vertices with no edges can be attached to the graph. For example, the preferential attachment model of Barabasi and Albert [7] includes the minimum degree of vertices as a central requirement for generating different power law approximations. In order to solve this issue we include a step that ensures that all edges have a degree of at least min_k. Unless otherwise stated, this value is set to 1.

4.2.3 Relationship with uniform random graphs

The uniform random Erdős-Renyi-Gilbert graphs discussed in chapter 2.5, have a natural fixed density form that equates to uniform sampling without replacement from the set of edges. This gives Algorithm 2 an interpretation as a weighted fixed density random graph model.

Formally, the definition of the uniform distribution on the unit circle is defined as [174],

$$\alpha_i = f_u(\theta_i) = \frac{1}{2\pi}. \tag{4.10}$$

Algorithm 2 CiGRAM construction algorithm. Used for internal module construction in Algorithm 3.

procedure FillGraph(nodes n, edges m, positions θ, score function s)
    Initialise empty list E
    Initialise V to size n
    $\alpha = s(\theta)$ # Set node scores
    while $|E| < m$ **do**
        β = Normalise(α)
        m_i = Sample($m - |E|, \beta$) # sample with replacement
        for $i \in V$ **do**
            Initialise S_i
            for $j \in V$ **do**
                if (i, j) in E or $i == j$ **then**
                    $S_{ij} = 0$
                else
                    $S_{ij} = \hat{\beta}(j|i)$ # See eq 4.9
            Normalise(S_i)
            V_s = SampleWRS(V, m_i, S_i) # sample without replacement
            for $j \in V_s$ **do**
                E.append((i, j)) # Add edge to graph
        for $i \in V$ **do**
            if $k_i = n - 1$ **then** # node has maximal degree
                $\alpha_i = 0.0$ # update weights so node cannot be selected
    return Graph $G(V, E)$

The distribution of α_i can then be normalised,

$$\beta_i = \frac{\alpha_i}{\sum_{j \in V} \alpha_j} = \frac{\frac{1}{2\pi}}{\sum_{j \in V} \frac{1}{2\pi}} = \frac{1}{n}, \tag{4.11}$$

which is equivalent to the probability of the uniform random graph.

In the limit $\sigma \to \infty$ the circular gaussian model is equivalent to the uniform distribution. Expressing the wrapped normal pdf in terms of a Fourier series expansion, we have a more convenient definition [174],

$$g(\theta; \mu, \varsigma) = \frac{1}{2\pi}\{1 + 2\sum_{p=1}^{\infty} \varsigma^{p^2} \cos p(\theta - \mu)\}, \tag{4.12}$$

where $\varsigma = e^{-\frac{\sigma^2}{2}}$. Given this definition, we can see that in the limiting case $\sigma \to \infty$ consequently, $\varsigma \to 0$. In the limit $\sigma \to \infty$, the term $\sum_{p=1}^{\infty} \varsigma^{p^2} \cos p(\theta - \mu)$ goes to 0 giving the final result,

$$g(\theta; \mu, 0) = \frac{1}{2\pi} = f_u(\theta). \tag{4.13}$$

This is equivalent to the uniform circular distribution in Equation 4.10.

4.2.4 Assortative configurations

The process described in Section 4.2.2 only really requires weights and the position variables could simply be replaced with some weighting function (in a similar vein to the Chung Lu model [115]). However, the use of latent variables is crucial for CiGRAM's ability to generate assortative and disassortative graphs. This process works by the inclusion of an additional parameter a that determines the propensity for nodes to connect, or not connect, according to the distance between points on the unit circle. This requires a single change to the secondary selection process, and Equation 4.9 now becomes,

$$\hat{\beta}(j|i) = \frac{\alpha_j e^{-a\delta(\theta_i, \theta_j)}}{\sum_{u \in \tau(i)} \alpha_u e^{-a\delta(\theta_i, \theta_u)}}, \tag{4.14}$$

where $\delta(\theta_i, \theta_j) = \frac{1}{\pi}(||\theta_i| - |\theta_j||)$, the radial distance between the vertices hidden variables. The reader should note the use of the absolute form of the variable $|\theta_i|$, forcing positivity. Without this constraint the positions $\theta_i = \frac{1}{2}\pi$ and $\theta_j = -\frac{1}{2}\pi$ have the maximal distance $|\theta_i - \theta_j| = \pi$ despite having the same score α_i, and therefore the same expected degree. This formulation equivalently

implies that the distribution of θ is concentrated on the unit semi-circle. Where $a > 0$, nodes of a similar degree have an increased propensity for connection. Where $a < 0$, nodes of a different degree have an increased propensity for connection. In the case of $a = 0$, Equation 4.14 is identical to Equation 4.9 as $e^0 = 1$.

The reasoning behind the use of radial distances is twofold. Firstly, as θ_i relates to a position in space that equates to the resulting node degree, determined by Equation 4.5, when a is positive, the closer θ_i and θ_j are to one another the more likely a connection is to form. Thus, nodes of a similar α score (and therefore degree) are more likely to form connections. Likewise, a negative value of a increases the propensity of nodes of a different degree to be connected.

The second aspect of this justification is that the Pearson correlation coefficient (PCC), used to measure assortativity r in Equation 2.26, has a geometric interpretation in the cosine similarity (CS) measure. The reader is reminded of the general definition for the PCC in Equation 3.1. The cosine similarity measure [176] of two vectors of the same length is defined as

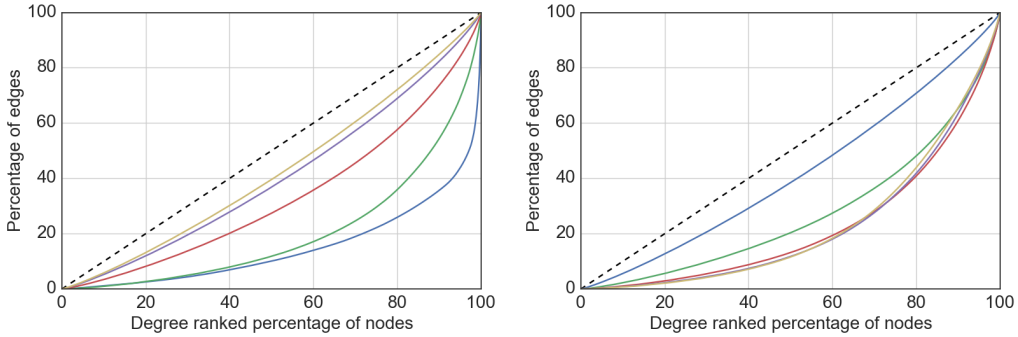
$$CS(x, y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}}. \tag{4.15}$$

This is simply the dot product of two vectors scaled by the product of their magnitudes. The difference between CS and PCC is simply the subtraction of the mean, $\bar{x}$ and $\bar{y}$, of each vector. Therefore $PCC(x, y) = CS(x - \bar{x}, y - \bar{y})$, giving us a geometric interpretation of the degree distribution and the resulting degree-degree correlations found.

4.2.5 Model results

This section highlights the topological properties that the above model is capable and incapable of fitting. One core aspect of the generated topology is the impact that node positions and scores have upon the resulting topology of networks. By fixing σ_s and σ_f in Equations 4.3 and 4.4, it is possible to observe the degree heterogeneity in the resulting networks. The simplest way to visualise this influence is through the use of Lorenz curves, used previously to show the influence of the wrapped Gaussian functions on the weights used

to generate the networks. Figure 4.3 shows that the two parameters have the opposite impact upon the degree distribution. Whilst σ_s increases the degree of nodes positioned closer to the pole π, σ_f impacts the number of vertices that will appear at each position. Fundamentally, neither varying position or score alone is enough to generate the range of weights required to generate extreme heterogeneity. Because the process uses sampling without replacement and a fixed minimum number of edges are used for each graph, the results of Figures 4.3 and 4.2 are very different.



(a) Varying σ_s *(b) Varying σ_f*

Figure 4.3: Influence of the model parameters σ_s (a) and σ_f (b) on the resulting degree distributions of the generated graphs. In Figure (a) σ_f is fixed at 0.8 and σ_s is set at 0.5 *(blue),* 0.875 *(green)* 1.25 *(red)* 1.625 *(purple) and* 2.0 *(yellow). In Figure (a) σ_s is fixed at 1.0 and σ_f is set at* 0.2 *(blue),* 0.65 *(green)* 1.1 *(red)* 1.55 *(purple) and* 2.0 *(yellow). All networks have $n = 2000$ and $d = 0.01$ with fixed $a = 0.0$.*

Figure 4.4 illustrates how the assortativity parameter, a, impacts the topological properties of a model with other parameters fixed. It was found that $a \in [-5, 5]$ is able to smoothly control the level of degree assortativity. Of note is that extremely heterogeneous graph configurations (coloured red) show little change in the degree assortativity in response to increased values of a, indicating a strong dependency between disassortativity and skewed degree distributions. The mean clustering coefficient (Equation 2.4) and modularity (Equation 2.11) of the generated networks also appear stable in response to a. It is important to note, however, that assortative connections do change the level of dependency between vertices, resulting in impact on other topological aspects of the network. The increase in modularity in response to the higher

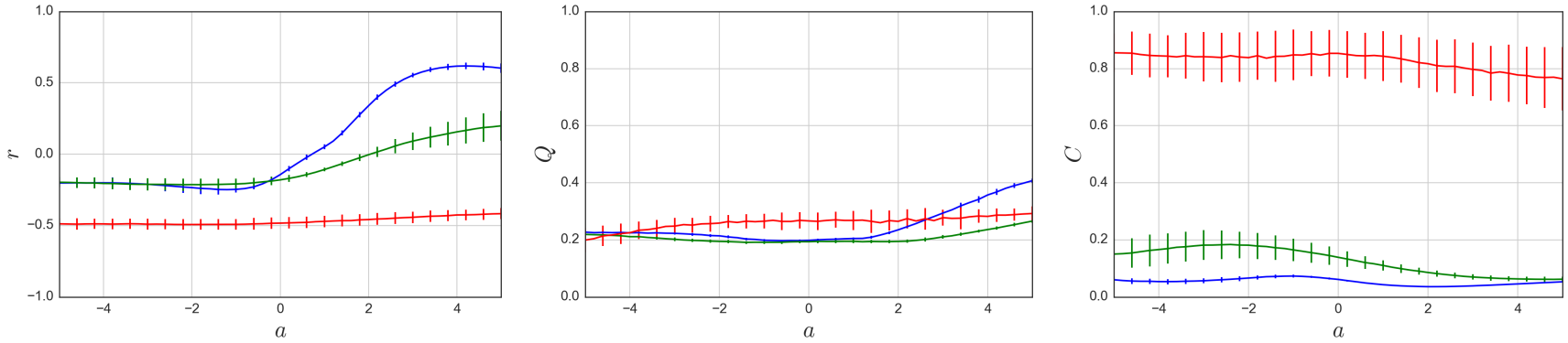
levels of assortativity is also potentially accounted for due to the nature of null model in Equation 2.11. The null model used does not include any dependency for degree-degree correlations, which will impact the probability of connections.

The a parameter also has a strong impact upon the degree distribution of the resulting network. This is demonstrated in Figure 4.5 which shows the complementary cumulative degree distribution of networks with fixed σ_s and σ_f with varying levels of a. In this sense, one cannot consider any of the parameters σ_s, σ_f, and a to be independent of one another. Selection of parameters that represent graph topology is an issue covered in Chapter 5.

One interesting aspect of assortativity is that it appears to be a property only measurable in sparse graphs. Figure 4.6 shows that as the density of the graph increases, the influence of the a parameter becomes negligible. For example, where $a = 3.0$, between $d = 0.01$ and $d = 0.1$, the assortativity drops to -0.4, despite having an increased dependency between vertices of the same degree. Indeed, graphs with an average degree above $\hat{k} \approx 30$ appear to have no positive assortativity, regardless of the the level of a. This does not conclusively prove that positive assortativity is necessarily a product of sparse graphs and it may be a limitation of the model. However, edge density places major constraints on other aspects of network topology, such as degree heterogeneity, with scale-free networks only being observable below a low density threshold [10]. An interesting aspect of this result is its implication for community structure. Internally, communities are very dense graphs, if a graph is highly assortative it may require a significant fraction of edges to be between communities or that the communities themselves are sparser than in other configurations.

4.2.6 Single community model summary

This section has presented the basis of the CiGRAM model; a fixed density non-modular random graph capable of generating heterogeneous degree distributions, disassortative, and assortative graphs. This is achieved by using a geometric approach to modelling the underlying probability space. The a parameter is shown to control the level of assortativity in the graph but it also has influence upon the degree distribution of the resulting network. None of



(a) Degree assortativity coefficient

(b) Modularity

(c) Clustering coefficient

Figure 4.4: Influence of parameter, a, on assortativity (sub figure a), modularity (sub figure b) and clustering coefficient (sub figure c). For graphs with $n = 2000$ *and* $d = 0.005$*, we tested three different degree distributions:* $\sigma_f = 1, \sigma_s = 1$ *(blue),* $\sigma_f = 0.5, \sigma_s = 0.8$ *(green) and* $\sigma_f = 0.8, \sigma_s = 0.5$ *(red). The last (red) is an extremely heterogeneous configuration within which a has no impact on assortativity. Each point is a mean of 100 samples. Error bars denote standard deviation.*

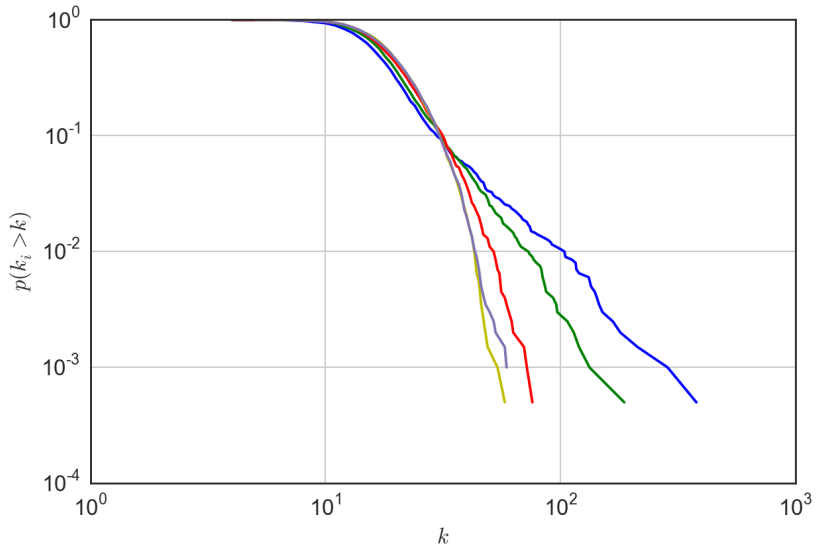


Figure 4.5: Influence of assortativity parameter on degree distributions. Networks are generated with fixed parameters $n = 2000$, $d = 0.01$, $\sigma_f = 0.8$, $\sigma_s = 1.5$. *a varies at levels* -2.0 *(blue)* -1.0 *(green),* 0.0 *(red),* 1.0 *(yellow)* 2.0 *(purple).*

the parameters discussed in this section allow modification of the clustering coefficient, indicating that transitive connections require increased dependency between vertices. Assortativity appears to be strongly related to the sparseness of the graph, with denser graph configurations showing no positive assortativity despite high levels of the a parameter. The next section moves towards the generation of modular graphs.

4.3 Graphs with modular structure

The following section explains the process of designing modular graphs. This approach is similar to a block model [177], however, there are several core differences. The densities for inter and intra module connections are fixed, rather than being governed by a specific parametrised probability. Furthermore, when edges are drawn, each community is treated as an isolated subgraph as are the edges between communities. More details regarding the difference between CiGRAM and stochastic block models are given in Section 4.4.

The explanation of the modular version of CiGRAM is divided into three subsections. Section 4.3.1 discusses the core assumption made about modules, that they are indistinguishable from random graphs, giving the motivation behind

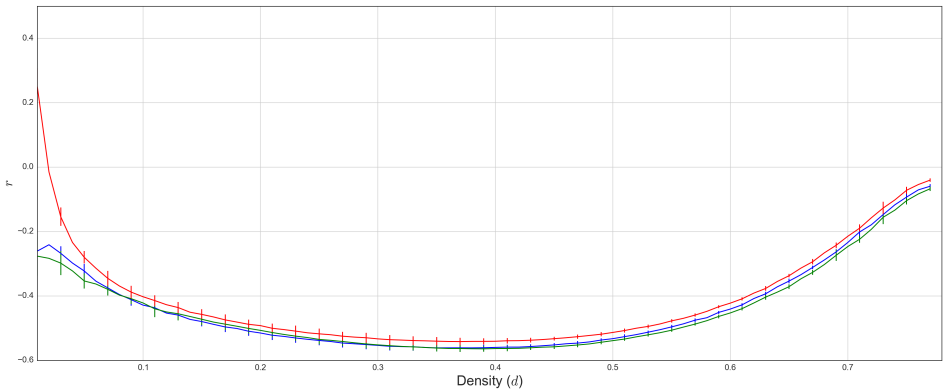


Figure 4.6: Dependency between assortativity and density. Three different values of a are used: 0.0 *(blue),* -3.0 *(green) and* 3.0 *(red) for* $n = 1000, \sigma_f = 0.8, \sigma_s = 0.8, k = 1$. *Each point is a mean of 100 samples. Error bars indicate standard deviation.*

Parameter	**Description**
K	Number of modules
σ_f	Node position variance
σ_s	Node score variance
a	Assortativity modifier
e_k	Fraction of edges between communities
p_o	Overlap probability modifier
$\tilde{\sigma}_f$	module position variance
$\tilde{\sigma}_s$	module score variance

Table 4.1: Description of CiGRAM parameters.

the choices made in subsequent sections. Section 4.3.2 includes explanation of a basic, uniform block structure in which K modules are defined and connected. In Chapter 3, in agreement with much of the literature [8], it was shown that modules in real networks are far from uniform in size. Section 4.3.3 defines the approach to modelling heterogeneous communities taking a similar approach to the wrapped Gaussian model described above. Under this configuration, nodes can also be members of multiple communities. A brief summary of all parameters is shown in Table 4.1.

4.3.1 Core assumption about modules

The most generally accepted assumption around the notion of modular community structure is that a community is a group of nodes that are more densely connected internally than externally [8]. In the following section, we investigate this implication with fixed density block graphs and refine the definition based on the notions of divisibility and connected groupings.

The definition of a sub set of nodes more densely connected internally than externally has lead to several planted partition models such as the LFR benchmark [9]. In this approach each node is given a fixed degree and a fraction of edges inside one or more specified blocks. In the formulation presented here, we take a different approach to the construction of communities, based on the assumption that a null model random graph lacks any modular structure.

More formally, this approach focuses on a simple question: assuming no information about external connections, should a given subgraph be considered a single community or not? When one approaches this problem, a different definition of a module becomes apparent. The conclusion drawn here is that a module is *any group of nodes that cannot be meaningfully divided into subgroups*. A random graph, either uniform or with a fixed degree distribution, does not include an increased probability for subsets of vertices to become connected (i.e. there is no dependence between connections). A *meaningful* division into subgroups must include a significantly higher dependency for subsets of vertices to become connected. In the model presented above, if two nodes have identical weights they are equally likely to form an edge with any other third node in the graph. In this formation, modular structure can only occur if nodes have an increased dependency of being connected.

This definition makes no statement about the *detectability* of a module. Indeed, it may be the case that the number of edges outside of the induced subgraph is significantly greater than those inside. In this case, a global detection algorithm would have great difficulty uncovering any such subgraph. This leads to a second assumption, that a subgraph module is only detectable within a wider graph if the vertices have a significantly higher probability of being internally connected than with the surrounding graph. This definition

matches the assumption that is widely used in the literature [8], but the reader should note that the definition of a community and any judgements about its detectability are not the same.

In Section 4.3, we use this definition by generating collections of fixed density subgraphs. This approach makes a simple specific prediction; when the internal density of communities is higher than the density connecting the groups, the network will have a significantly higher level of localised density (clustering coefficient) due to the increased dependency between vertices. In other words, where there is an increased probability that subsets of vertices will be connected, the average clustering coefficient of the network will be higher than in the null case. For this reason, CiGRAM model does not allow direct control of the clustering coefficient as this may interfere with any resulting modular structure.

4.3.2 Basic block structure

The simplest way to generate a block structure is to generate a graph with K uniformly sized blocks, giving the set of communities $\mathcal{C}$. In this simplified definition, each block contains $\frac{n}{K}$ nodes and $\frac{m}{K}$ edges. Internally, a graph is connected in an identical manner as described in Algorithm 2.

The θ_i for each node are considered a global property. Initial experimentation, with results not presented in this thesis, was conducted having a different position variable for inter and intra community connections, however it made little quantifiable difference. Furthermore, any increase in the overall model complexity must be fully justifiable in terms of the topology that can be generated. Having multiple θ_i is not parsimonious, given it was found to have little influence on the resulting network.

The number of inter-community edges is determined by the parameter $e_k \in [0, 1]$. This also determines the intra community edge density, or the fraction of edges inside communities. The maximum number of connections between communities is the sum over the cardinality of the Cartesian product of all pairs of communities $\sum_{c_l, c_k \in \mathcal{C}} |c_l \times c_k|$. However, in the case of most sparse configurations this upper density limit cannot be achieved. As a consequence,

where $e_k = 1.0$, the resulting network becomes a K-partite graph, and all edges must exist between communities. Under this model, edges between communities are assigned in an identical manner to the single community model, with the added condition that edges cannot be assigned to nodes in the same community. This is distinct from stochastic block models which define a probability for each pair of communities being connected.

4.3.3 Heterogeneous modules with overlapping nodes

This section describes the approach to generating modular graphs with CiGRAM. The pseudo code in Algorithm 3 outlines how this process is completed. The following subsections elaborate on this algorithm.

Figure 4.7 shows a visual example of a modular graph constructed with CiGRAM, showing the geometric nature of the communities allowing assortative structure. The Geometric positions in this figure relate to the latent positions used to determine the probabilities for constructing edges.

Generating modules

This section discusses lines 1 to 20 of Algorithm 3. In order to match the heterogeneous community structure observed in Chapter 3, as well as results found in the literature [178], the size and density of communities must be configurable over a range of scales. In order to achieve this goal, the method described here uses the same dual wrapped Gaussian distribution approach taken above. Formally, each community has a position $\tilde{\theta}_k$ drawn from a wrapped normal distribution with standard deviation $\tilde{\sigma}_s$ and a mode of $\mu = 0$. The weight for each community $c_k \in \mathcal{C}$ is then defined as

$$\tilde{\beta}_k = \frac{\tilde{\alpha}_k}{\sum_{l \in \mathcal{C}} \tilde{\alpha}_l}, \tag{4.16}$$

where $\tilde{\beta}_k$ gives the probability that a given node will be selected to be a member of module k. Before the edge density can be assigned, modules must be assigned nodes. Each node must be a member of at least a single module; this is determined by the weighted random selection with probabilities $\tilde{\beta}_k$. Where this is desired, the minimum number of nodes is assigned to each community before the sampling procedure is completed.

Algorithm 3 CiGRAM Modular random graph construction

procedure ModularGraph(nodes n, edges m, between edges e_k, overlap p_o, modules K, node positions θ, module positions $\tilde{\theta}$, score function s, module score function $\tilde{s}$)
  Initialise empty list E
  Initialise V to size n
  $\alpha = s(\theta)$ # Set node scores
  Initialise K empty modules $\mathcal{C}$
  SetCommunityNodes(n, $\mathcal{C}$, $\tilde{\theta}$, p_o)
  # Assignment of edge count inside each community
  m_k = SetCommunityEdgeCount(n, $\mathcal{C}$, $\tilde{\theta}$, e_k * m)
  set ReassignCount = 0
  for Module $c \in \mathcal{C}$ **do**
  # Assignment of edges inside each community, Algorithm 2
    SubGraph = FillGraph(c, m_k, θ, s)
    for Edge $e \in SubGraph$ **do**
      if $e \in E$ **then** # Edge already exists, must be reassigned
        ++ReassignCount
      else
        E.append((i, j)) # Add edge to graph
  while ReassignCount > 0 **do**
    c = SelectModule($\mathcal{C}$, $\tilde{\theta}$) # Use Algorithm 2 to assign extra edge
    AddInnerEdge(c)
    --ReassignCount
  while $|E| < m$ **do** # Assignment of edges between communities
    Set $\tilde{\beta}_k = \frac{\tilde{\alpha}_k}{\sum_{l \in \mathcal{C}} \tilde{\alpha}_l}$
    Select c = SelectFirstModule($\tilde{\beta}_k$)
    Select i = SelectNodeFromModule(c)
    Select j with $\hat{\beta}(j|i)$ # See Equation 4.19
    E.append((i, j))
  return Graph $G(V, E)$, modules $\mathcal{C}$

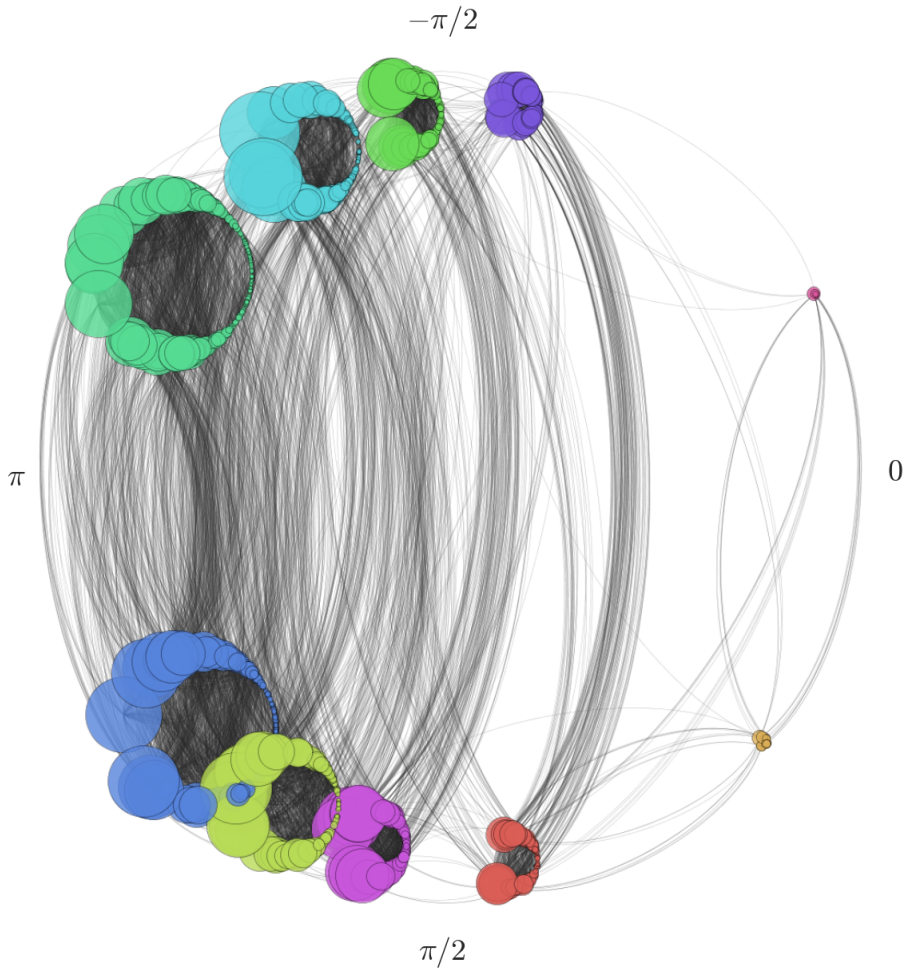


Figure 4.7: Assortative, heterogeneous community structure generated by the model with $n = 1000, d = 0.03, K = 10, \sigma_f = 1.1, \sigma_s = 0.9, \tilde{\sigma}_f = 1.6, \tilde{\sigma}_s = 1.0, a = 4.0, e_k = 0.1, p_o = 0.0$. *The community positions are derived from the wrapped Gaussian density function (see Equation 4.1) similarly to the positions of nodes within each community. The nodes belonging to each community are marked with a distinct colour. Node size is proportional to its degree.*

In order to allow overlapping modules, p_o acts as a probability modifier for subsequent module selection. A node can be a member of any number of modules and, in principle, p_o can take on any real value. However, as subsequent results will show, even a moderate level p_o can result in extreme levels of overlap. For each additional community $\mathcal{C}_l$, a node i becomes its member with probability

$$Pr(c_l \in \mathcal{C}_l | c_k \in \mathcal{C}_k) = p_o \frac{\tilde{\beta}_l e^{-a\delta(\tilde{\theta}_k, \tilde{\theta}_l)}}{\sum_{u \neq k} \tilde{\beta}_u e^{-a\delta(\tilde{\theta}_k, \tilde{\theta}_u)}}, \tag{4.17}$$

where c_k refers to the first module in which a node is assigned. We then refer to the set of communities a node i is a member of as M_i. The assortativity parameter and the use of module size is to allow a controllable level of assortativity. If a node is positioned within a large module, its potential degree is significantly higher than one positioned in a smaller module. As a consequence, the position of each module is used to determine the probability of overlap between modules. A similar approach is used to model the edges between communities, explained later in this section.

Given that nodes are assigned to one or more modules, the number of edges for each module can be assigned. This process is determined in the same fashion as the node selection process. $\tilde{\beta}_k$ is used for weighted sampling with replacement from the $m(1 - e_k)$ edges available inside each community. The internal structure of each community is then generated in an identical fashion to the process described in Algorithm 2, the σ_f, σ_s and a parameters are identical for the internal structure of all communities.

The overlapping nature of this community structure can create a challenge with multiple edges assigned between the same pair of nodes. As each community is generated independently, this occurs when two nodes are members of the same group. Consequently, edges must be reassigned when this occurs to ensure that the desired edge density is achieved. Here, the assignment is completed in the same manner as described in Algorithm 2, with the exception that adjacency assigned in other modules is known, meaning that the processes are no longer independent.

Generating edges between modules

This section discusses lines 22 to 27 of Algorithm 3. The remaining edges in the resulting network are assigned between communities. To add an edge, we select a community $\mathcal{C}_k$ with probability $\tilde{\beta}_k$ and a node $i \in \mathcal{C}_k$ with probability β_i. A second node from the community $\mathcal{C}_l$ is selected using a modified form of the matrix described in Equation 4.8,

$$S_{ij} = \begin{cases} 0 & \text{if } |M_i \cap M_j| \neq \emptyset \text{ or } A_{ij} = 1, \\ \hat{\beta}(j|i) & \text{otherwise} \end{cases} \tag{4.18}$$

where $\hat{\beta}(j|i)$ is now defined as

$$\hat{\beta}(j|i) = \frac{\gamma(M_i, M_j)\alpha_j e^{-a(\delta(\theta_i,\theta_j)+\delta(M_i,M_j))}}{\sum_{u\in\tau(i)} \gamma(M_i, M_u)\alpha_u e^{-a(\delta(\theta_i,\theta_u)+\delta(M_i,M_u))}}, \tag{4.19}$$

where $\tau(i)$ is the set of vertices not in the same module as i and not adjacent to i, and the distance between communities which nodes i and j are members of is

$$\delta(M_i, M_j) = |\max_{k\in M_i}\{|\tilde{\theta}_k|\} - \max_{l\in M_j}\{|\tilde{\theta}_l|\}|, \tag{4.20}$$

with the inter community connectivity $\gamma(M_i, M_j)$ defined as

$$\gamma(M_i, M_j) = \sum_{\{(k,l)\in M_i\times M_j | k\neq l\}} \tilde{\alpha}_k\tilde{\alpha}_l e^{-a\delta(\tilde{\theta}_k,\tilde{\theta}_l)}. \tag{4.21}$$

The strength of connection between two nodes in different communities depends on γ. Thus, communities of similar size are more likely to be connected when $a > 0$ and communities of different sizes are more likely to be connected when $a < 0$. Furthermore, the fact that we take into account the distance between the communities (see Equation 4.20), makes it unlikely that high degree nodes will connect to small communities when assortativity is desired. Whilst this process is necessary to generate networks with assortative structure, subsequent sections will show that the density of the graph and the internal density of the communities makes this process extremely flexible from a modelling perspective.

4.3.4 Modular model results and discussion

Figure 4.8 shows how the underlying community structure changes the node degree distribution. As K increases the number of nodes with high degrees

drops and more nodes with low degree appear. The exact reason for this is difficult to quantify, though it is likely an effect of increased node separation into smaller and smaller communities than cannot be compensated by the limited number of inter-community connections. This is a major limitation of CiGRAM when compared to fixed degree models. However, given the random nature of graphs it should be expected that dense, modular structure has a strong impact upon the degree distribution just as heavy tailed degree distributions cannot be ignored when considering modular structure [130].

In a result related to Figure 4.6, Figure 4.9 shows that this heterogeneity leads to increased assortativity. Each cluster corresponds to a different level of variation in community edge density, resulting from the selection of the community position and score function parameters $\tilde{\sigma}_f$ and $\tilde{\sigma}_s$. Formally, we define the density of a community as a density of the subgraph induced on the nodes which belong to this community. The standard deviation of the community density is defined as

$$\sigma_{dc} = \sqrt{\frac{\sum_{c_k \in \mathcal{C}} (d(c_k) - \mu_d)^2}{K}}, \tag{4.22}$$

where $\mu_d = \frac{1}{K} \sum_{c_k \in \mathcal{C}} d(c_k)$ is the mean community density.

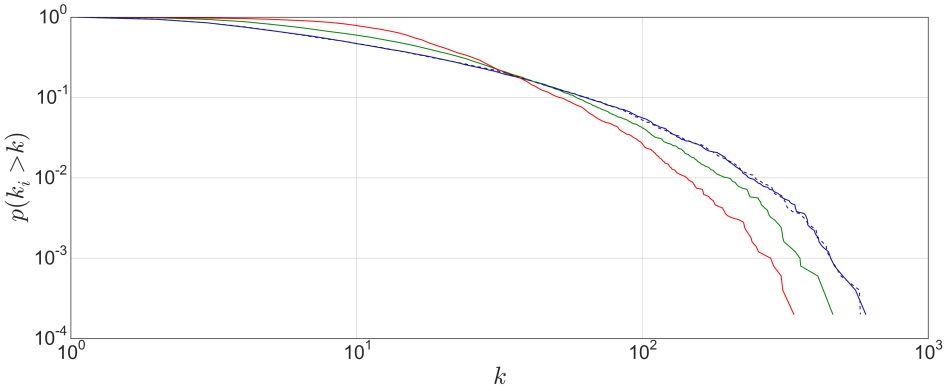


Figure 4.8: Dependency between the number of communities and degree distribution. Three values of k are used; 2 (blue), 50 (green) and 300 (red) for $n = 5000, \sigma_f = 0.8, \sigma_s = 0.8, \tilde{\sigma}_f = 0.8, \tilde{\sigma}_s = 0.8, d = 0.005, e_k = 0.1$. Results for $k = 1$ are shown in grey dashed lines.

The implication of this result is that in addition to limits imposed by the overall graph density discussed earlier, the degree assortativity is also strongly influenced by the variation in the density of individual communities.

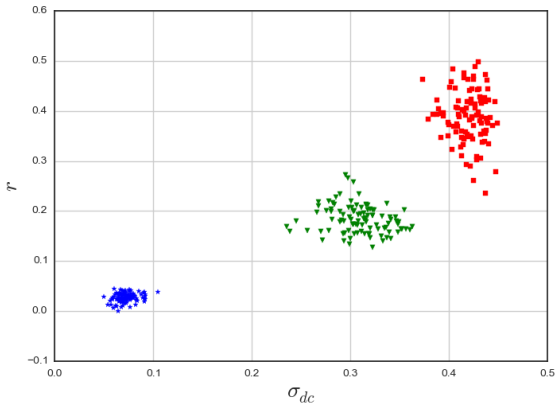
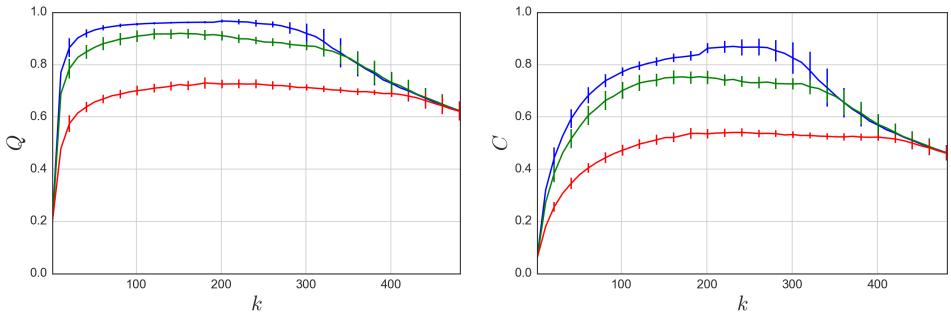


Figure 4.9: Dependency between assortativity and the standard deviation of community density (σ_{dc}). Three variants of community position and score parameters are used: $\tilde{\sigma}_f = 1.8$ $\tilde{\sigma}_s = 1.8$ (blue), $\tilde{\sigma}_f = 1.1$ $\tilde{\sigma}_s = 1.1$ (green), $\tilde{\sigma}_f = 0.7$ $\tilde{\sigma}_s = 0.7$ (red). We show 100 samples with fixed model parameters $n = 5000, k = 100, \sigma_f = 1.0, \sigma_s = 0.8, d = 0.0015, e_k = 0.1$.

Degree assortativity depends on a combination of both global and local network properties.

The interdependence between community structure and other topological properties is highlighted in Figure 4.10. The modularity of the network increases with K, as does the clustering coefficient. Their growth rate appears to be bounded by the fraction of edges between communities (e_k). By necessity, as K increases the size of each community decreases, resulting in more dense subgraphs. When K becomes too large for the number of nodes involved, the clustering coefficient and modularity begin to decline. It appears that for K above $n/10$ the average clustering coefficient begins to fall, though results depend heavily upon the size and density of the communities, as well as the value of e_k. This demonstrates a strong relationship between the community structure, as defined here, and high clustering coefficients. This is, perhaps, a natural relationship given that modules can be considered areas of localised density in otherwise sparse graphs. For this reason, we argue that networks with a predefined ground truth community structure should not have a fixed level of transitivity.

The influence of the community overlap on the modularity and clustering coefficient is shown in Figure 4.11. Both modularity and clustering coefficient are



(a) Modularity *(b) Clustering*

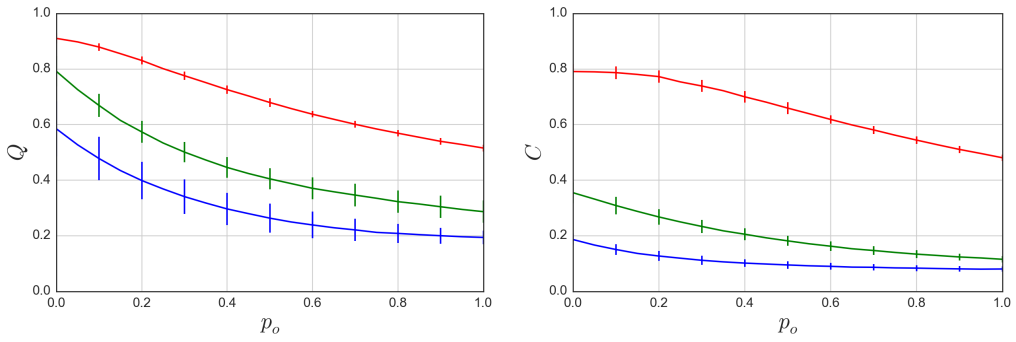
Figure 4.10: Modularity and clustering with increasing number of communities K. Both appear to be bounded by the number of edges between communities e_k varies between 0.01 *(blue),* 0.1 *(green) and* 0.3 *(red) for model* $n = 2000, d = 0.01, \sigma_f = 1, \sigma_s = 1, a = 0.05$. *Each point is a mean of 100 samples. Error bars show standard deviation.*

strongly reduced when the overlap level increases. As the separation between the communities gets weaker due to increased number of inter-community connections introduced by each multi-community node, the network structure gets closer to the null model assumed in modularity definition (see Equation 2.11). Furthermore, if a node belongs to multiple communities, it is less likely that its neighbours will become connected to form a triangle. As a result, the clustering coefficient drops due to the constraint on the number of connections between communities.

Figure 4.12 visually shows how the increasing level of overlap changes the structure of a network. At $p_o = 0.2$, the communities become mixed, but the overall structure of the graph is not significantly impacted. At $p_o = 0.8$, the graph is largely indistinguishable from a single community model.

4.3.5 Modular graph summary

This section has described the CiGRAM modular random graph generator. This model is based on an underlying assumption that non-modular random graphs are equivalent to modules. The model, therefore, creates a ground-truth community structure with a configurable level of overlap and inter community edges. This approach is shown to create higher levels of clustering and assortativity than are possible in non-modular configurations. A high level of overlap



(a) Modularity *(b) Clustering*

Figure 4.11: Modularity and clustering for increasing p_o. k varies between 5 *(blue),* 15 *(green) and* 300 *(red) for model* $n = 2000, d = 0.01, \sigma_f = 1, \sigma_s = 1, a = 0.05, e_k = 0.05$. *Each point is a mean of 100 samples. Error bars show standard deviation.*

is shown to create networks with less modular structure, closer to non-modular graphs. Furthermore, the sparseness of the underlying community structure is shown to strongly relate the level of degree assortativity in networks.

4.4 Related topological models

There are several approaches to modelling network structure closely related to CiGRAM. This section discusses geometric models [126, 173] and stochastic block models [177], most notably degree corrected stochastic block models [130]. In the domain of graphs designed to construct testable modular structure, we compare CiGRAM to the popular benchmark graphs in Chapter 6.

The geometric approach of [126] is used for link prediction in complex biological networks. Papadopoulos et al. [126] uses latent geometric positions in hyperbolic space to determine the connections between nodes. One angle is used to determine the similarity between vertices, and another to determine their popularity. The probability of nodes connecting is then a product of the distances in this space. The work of Serrano et al. [173] uses a very similar approach to [126], however in this case the angles are of a circle geometry rather than a hyperboloid. In this approach, the position of verticies is uncovered using Markov Chain Monte Carlo simulation of points on the unit circle. Again, the distances between the nodes determine the probability of vertices connecting.

CiGRAM does not use distances in space to determine the vast majority of edges. Indeed, CiGRAM allows nodes that are close in space to be less likely to form a connection. Furthermore, if the assortativity modifier $a = 0$, then the distances have no impact upon the topology of the network. The objective of CiGRAM is to model the heterogeneous probability space of networks with geometry; it should not be assumed that the underlying positions relate to any real geometry.

The degree stochastic block model [130] uses a similar block based approach to modelling modular structure. Block models may be used for both inferring underlying modules and generating topology. As with CiGRAM, a node is assigned to one of K blocks. Each pair of blocks is connected with a specified probability as a parameter, which is not the case in CiGRAM. In the degree corrected stochastic block model, a parameter is required for each node of the network which influences its degree. Each community is connected, internally, with a uniform probability; an assumption not made by CiGRAM. Furthermore, CiGRAM does not require parameters for determining the probability of edges connecting between blocks as this is simply a product of the positions in space.

4.5 Implementation and performance

The model is implemented in Python with the use of the networkx library [179] for ease of use. The network generation process is computed with a C++ extension, allowing the faster generation of large scale graphs. Software is available for GNU/Linux distributions at `http://cigram.ico2s.org`.

Figure 4.13 (a) demonstrates a quadratic-time dependency on the size of the graph (n). Figure 4.13 (b) shows a linear-time dependency on the number of edges (m). The time complexity of the generator in Algorithm 2 is $O(mn^2)$. As the maximum number of edges is $\frac{n(n-1)}{2}$, this can be restated as $O(n^4)$ in the worst case of fully dense graphs. The generation time decreases with the number of communities (see Figure 4.13 (c)). The graph generation performance was measured on an Intel Sandybridge E5-2670 2.6GHz processor with 128 GB of RAM.

4.6 Chapter summary

This chapter has introduced CiGRAM, an approach to generating synthetic networks with the heterogeneous degree distributions and degree-degree correlations found in real world networks. This stems from a geometric approach to modelling the underlying probabilities that determine graph properties, and follows as a natural extension of fixed density Erdős-Renyi-Gilbert uniform random graphs. In Chapter 5 this approach is shown to be able to fit many real world, heavy-tailed distributions. This is particularly notable as it is completed without the need for any growth or preferential attachment based mechanisms. An assumption that modular structure is indistinguishable from collections of random graphs allows the generation of clustering coefficients found in real world graphs. The difficulty of generating highly assortative dense graphs is highlighted, showing that positive degree-degree correlations may require sparse graphs with sparsely connected communities. This achieves one of the core objectives of this Thesis; to provide a ground-truth modular structure that allows the evaluation of module extraction algorithms. The generation of this realistic community structure allows the testing of module detection algorithms, a topic investigated in Chapter 6.

To summarise, the contributions of this Chapter are as follows:

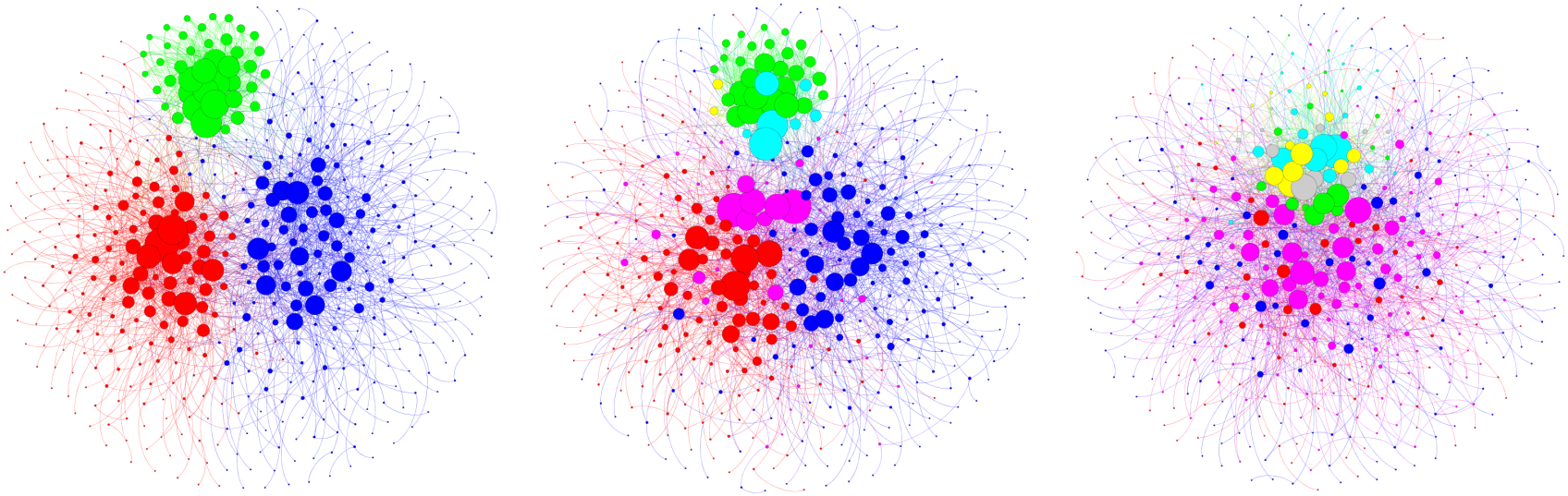
- The generation of heterogeneous degree distributions through use of wrapped Gaussian distributions.
- A construction algorithm for producing fixed density graphs through weighted sampling without replacement.
- The configuration of networks to create assortative structures through use of angular distances.
- A formal definition of community structure based on the assumption that non-modular random graphs can be considered modules.
- The construction of networks with a known, ground-truth modular structure by extending the basic model.

- The demonstration that high clustering coefficients emerge as a product of community structure.

- Indications that assortativity requires sparse graphs and the implications this has for community structure in networks.

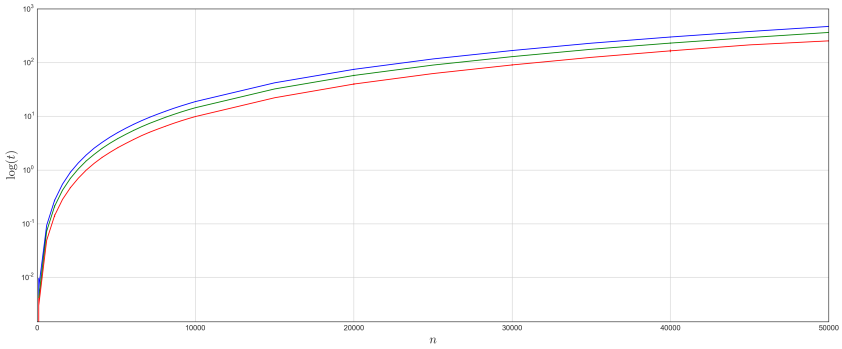
Fundamentally, this chapter answered the two research questions outlined in the introduction section. The generation of assortative topology is achievable through a geometric definition of the probability space. The formal definition of a module is that it is indistinguishable from a non-modular random graph of equivalent edge density.

The approach taken to modelling topology in this chapter is not without its limitations. Crucially, the performance of the generative network is limited by the nature of sampling without replacement. Furthermore, estimating the resulting topological properties of this model proved to be extremely challenging. Chapter 5 looks at approaches to overcoming this limitation by fitting real world graphs based on spectral distances and topological measures.

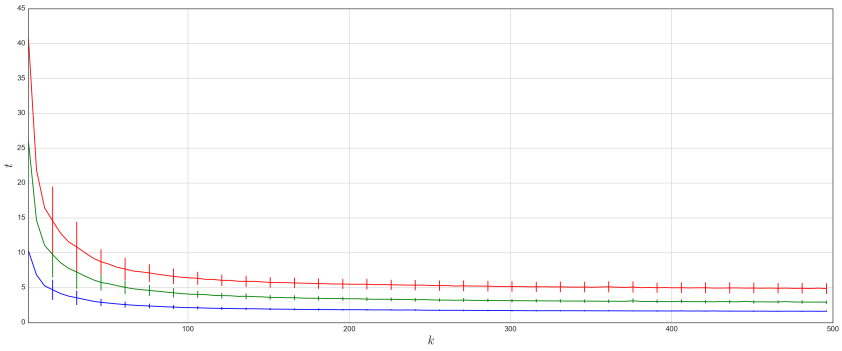


(a) $p_o = 0$ *(b)* $p_o = 0.2$ *(c)* $p_o = 0.8$

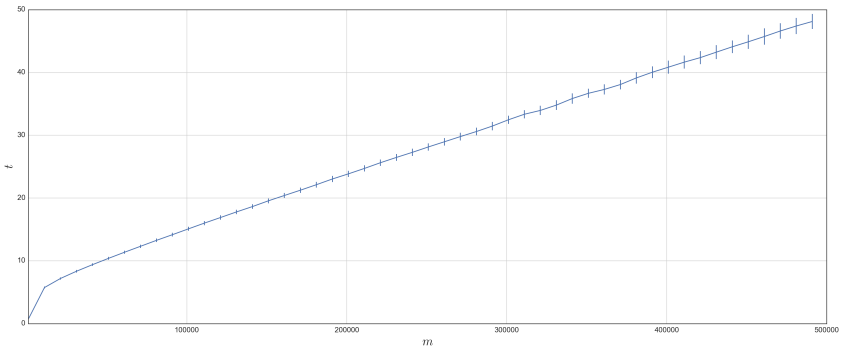
Figure 4.12: Example graphs generated with $n = 500, d = 0.015, \sigma_f = 1.2, \sigma_s = 0.9, a = 0, k = 3, e_k = 0.12$ *and varying levels of overlap,* p_o*. Node positions,* θ_i*, remain the same, but the overlap heavily alters the structure of the graph. Colour indicates node community, node size is proportional to its degree.*



(a) Number of nodes



(b) Communities



(c) Number of edges

Figure 4.13: Time (t) in seconds to generate graphs. ***(a)*** *Scaling with* n. $\sigma_f = 0.8, \sigma_s = 0.8, a = 1, k = 1$ *with variable density:* $\frac{10}{n}$ *(blue),* $\frac{20}{n}$ *(green),* $\frac{30}{n}$ *(red).* ***(b)*** *Scaling with communities.* $n = 10\ 000, \sigma_f = 0.8, \sigma_s = 0.8, a = 1$ *with variable density:* 0.001 *(blue),* 0.0045 *(green),* 0.008 *(red).* ***(c)*** *Scaling with edges* $n = 10\ 000, k = 1, \sigma_f = 0.8, \sigma_r = 0.8, a = 1.5$. *Each point represents the mean of 100 samples and the error bars show standard deviation.*

Chapter 5

Model parameter selection

5.1 Introduction

Chapter 4 presented CiGRAM and demonstrated its ability to generate graphs with properties found in real world networks such as heterogeneous degree distributions, configurable levels of degree assortativity and high clustering coefficients. A drawback of CiGRAM, however, is that none of these parameters are separable. For example, whilst increasing the a parameter results in higher levels of degree assortativity, r, this also has an impact on the degree distribution of the graph. This chapter presents a method for optimising the parameters of CiGRAM to fit real world complex networks, both biological and non-biological in nature. This chapter aims to answer the research question: can CiGRAM be fitted to empirical data? In order to achieve the core aim of this thesis, to provide a well grounded method for module detection algorithm selection, CiGRAM must be tuned to form an adequate model of real world datasets. In Chapter 6, the fitted models are used for the evaluation of algorithms.

A core problem related to this work is the difficulty in formally measuring the similarity between two large graphs. One distance measure that is used in some cases is the graph edit distance [180], which takes the minimum number of rewiring operations required to make one graph isomorphic to another. The graph edit distance is extremely costly to compute, being NP-Hard and scaling only to very small networks. Consequently, this chapter explores two alternative methods; spectral distances between the eigenvalues of the normalised Laplacian [181] and topological summary statistics in the form of

degree distributions, assortativity and clustering coefficients.

In terms of the spectral distances, the analysis is conducted on small metabolic networks. The Euclidean distance [182] and Jensen-Shannon divergence [183] are used to evaluate the distances between eigenvalue distributions of metabolic networks. This work also presents a novel distance measure in the form of the Kolmogorov-Smirnov distance between cumulative graph spectral distributions. CiGRAM parameters are then tuned to fit metabolic networks using the distance functions as a cost function in a particle swarm optimiser [184]. This method is found to have some significant limitations, such as the lack of scalability to large graphs and the inability to well represent the full network topology. In addition to the results in this chapter, the influence the parameters of CiGRAM have upon the Laplacian spectra of graphs is investigated in Appendix B, Section B.1.

An alternative approach based on using topological summary statistics is proposed, being capable of scaling to larger biological and non-biological networks. The use of a particle swarm optimisation is applied here and is shown to find parameters for CiGRAM which form a good match for the desired summary statistics of empirical networks. However, the final section of this chapter discusses the limitations of this approach with regards to its ability to generate all salient features of networks. This also includes a discussion of the properties that CiGRAM is capable of generating.

5.2 Particle Swarm Optimisation

This section describes the optimisation procedure used for fitting networks in later sections. The high dimensionality of CiGRAM and the lack of any known gradients in the search space make optimisation a challenge. Due to a lack of knowledge about the search space for any given graph, meta-heuristics are an appropriate form of optimisation strategy. The stochastic nature of CiGRAM also makes evaluation challenging, for this reason each fitness evaluation is performed on 5 candidate solutions. The number of candidates is a trade-off between higher accuracy for model fit and computation time. For the larger networks in this study, 5 evaluations was selected as it appears sufficient to

avoid extreme variation whilst still being able to compute a fit in reasonable time.

Particle Swarm Optimisation (PSO) [184] is a nature inspired meta-heuristic where a population of particles search the space of solutions following simple formulas of motion, based on current position and velocity. PSO has been shown to withstand noise in terms of fitness functions [185]. Parsopoulos and Vrahatis showed that introducing noise into the results of benchmark fitness functions not only had minimal impact upon the algorithm result, but also helped the swarm avoid getting trapped in local optima [186]. Whilst we cannot state, formally, that the nosie from CiGRAM is Gaussian in nature, there is certainly a degree of model error. The fact that the PSO procedure can adapt to such noise is helpful in exploring as large a region of the search space as possible.

Inspired by the notion of flocks of birds collectively foraging to find food, PSO works by iteratively moving the collective set of particles across a given search space. Each individual in the swarm is composed of three vectors, current position x_i, previous best fit position p_i and velocity v_i. The position vectors x_i and p_i should be seen as the position of the particles within the search space of the problem and therefore relate to the model parameters. The optimisation of the swarm depends not on the motion of individual particles but the communication between them. Analogous to a social network, each particle has a set of neighbours with which it communicates. There are various topologies that can be used to improve the process, such as small world networks [187], however, the standard approach of a ring topology, in which each particle has two neighbours, is used here. The best position vector observed in each particle's neighbourhood is referred to as l_i.

Each particle is first assigned a random position vector x_i and velocity v_i within the bounds of the problem. The population is then iteratively updated. At each iteration, each particle's current position x_i is evaluated. If $f(x_i) > l_i$ then l_i is updated to the current location, the same applies to p_i.

The velocity of the particle is then updated according to the following equation,

$$v_i^{t+1} = wv_i^t + \varphi_1 U_1^t(p_i^t - x_i^t) + \varphi_2 U_2^t(l_i^t - x_i^t) \tag{5.1}$$

where t refers to the current time step and $t+1$ refers to the next iteration, w as the inertia weight and, φ_1 and φ_2 are termed acceleration coefficient parameters. In this study, φ_1 and φ_2 are set to 2.1 and the inertia is set at $w = 0.5$. U_1 and U_2 are randomisation functions that multiply the vector elements in the interval $[0, 1)$. Selecting these parameters is somewhat arbitrary in nature, the decision made here was simply based on the understanding that much of the literature on PSO uses these values [185]. The position of the particle at the next iteration is determined entirely by the updated velocity,

$$x_i^{t+1} = x_i^t + v_i^{t+1}. \tag{5.2}$$

The PSO procedure continues to update until a maximum number of iterations is reached. For the purposes of the experiments conducted in this chapter, a population of 20 particles is used. Unless otherwise stated, the boundaries for the particle swarm optimisation process are set in the range $[0.3, 2.2]$ for $\sigma_f, \sigma_s, \tilde{\sigma}_f$ and $\tilde{\sigma}_s$ with $K \in [0, n/10]$ (the approximate point at which clustering began to drop in Chapter 4) with a specified based on the network in question.

5.3 Dataset descriptions

In order to test the ability of CiGRAM to fit real world data, a variety of datasets are used, these are highlighted in Table 5.1. The main focus of this thesis is biological data and so five biological datasets are used that fit into three categories, protein-protein interaction networks taken from Yeast [188] and Arabidopsis thaliana [4], the *Arabidopsis thaliana* correlation of expression network SeedNet [43] analysed in Chapter 3 and metabolic networks taken from *C elegans* [107] and *E coli* [54]. The model is also capable of generating topology observed in non-biological complex networks and so 4 additional datasets are analysed. This includes the Open Flights air transportation network, the US Power Grid, the PGP key signing network and a social network of Hamster owners.

The vertices of the Open Flights network are all cities that contain one or more airport and the edges relate to flights available between them. Note that the Open Flights network is technically directed, however, there is a high degree

of symmetry between the in and out degree distributions [189], consequently, the network is treated as an undirected graph.

The Hamster network is a social network for hamster owners taken from the website *hamsterster.com* (now defunct) made by the Konect project [190]. Vertices represent users and edges represent friendships.

The PGP network is a trust network between individuals that use the Pretty Good Privacy asymmetric encryption method [191]. Each vertex represents an individual and each edge indicates that the pair have signed each other's private keys.

The US Power Grid network [6] represents all the transformers (nodes) and power lines between them (edges) in the United States.

As these models are to form the basis of evaluations for module detection algorithms in Chapter 6, only the largest connected component is considered. The results of Chapter 4 give us some intuition about the parameters that real world graphs are likely to have. Specifically, the a parameter is set inside a range that matches the level of observed assortativity. These parameter ranges are highlighted in Table 5.1.

Category	Network	n	m	density	C	r	a_{min}	a_{max}
PPI	Yeast	2284	6646	0.0025	0.135	-0.099	-5.0	0.0
	Arabidopsis	4519	11096	0.0011	0.099	-0.197	-6.0	-1.0
Metabolic	*C elegans*	453	2025	0.0198	0.646	-0.226	-5.0	0.0
	E coli	294	730	0.0169	0.292	0.609	1.0	6.0
Co-expression	SeedNet	8485	501522	0.0139	0.502	0.177	0.0	5.0
Non-biological	Open Flights	2905	15645	0.0037	0.456	0.049	-2.0	3.0
	US Power Grid	4941	6594	0.0005	0.08	0.003	-3.0	2.0
	PGP	10680	24316	0.0004	0.266	0.238	1.0	6.0
	Hamster	1788	12476	0.0078	0.143	-0.089	-2.0	2.0

Table 5.1: Topology of datasets fitted with CiGRAM. The last columns a_{min} and a_{max} indicates the given range of assortativity parameters to be used in the optimisation procedure.

5.4 Fitting graph spectra

This section introduces the distance metrics used when fitting real world networks. For this purpose we compare three metrics, a Euclidean spectral

distance [182], the Jensen-Shannon distance [183] and a novel approach to fitting graph spectra in the form of the Kolmogorov-Smirnov distance. It is important to note that these metrics are, strictly speaking, *pseudo-metrics* in the sense that not all graph topology is captured. By ignoring the eigenvectors of the networks and only computing the distance between eigenvalues, they likely fail to capture the full structure of the network. Fundamentally, this means that two topologically non-isomorphic graphs can, in principle, have a spectral distance of 0.

Euclidean distance

Whilst a visual qualitative analysis of the spectra is useful, a more formal quantitative approach can be used through the use of distance metrics. One approach to computing the distance between spectra that has been used before [182] is the Euclidean distance of the eigenvalues of the normalised Laplacian. Formally, given two sets of ranked eigenvalues $\{\lambda_0 < \lambda_1 < ... < \lambda_n\}$ and $\{\mu_0 < \mu_1 < ... < \mu_n\}$ relating to graphs G and G', respectively such that $\sum_i \lambda_i^2 \leq \sum_i \mu_i^2$, one can measure the distance between them by,

$$D_e = \sqrt{\frac{\sum_i (\lambda_i - \mu_i)^2}{\sum_i \lambda_i^2}}. \tag{5.3}$$

One notable limitation of Equation 5.3 is that it requires the spectra to be of the same size. If the graphs in size differ by even a single node, the metric is not defined.

Jensen-Shannon distance

Banerjee [183] uses the Jensen-Shannon distance between graph spectra to compare the structural properties from networks of different organisms. Jensen-Shannon measure is a symmetric from of the Kullback-Leibler divergence measure which defines the difference between two probability distributions p_1, p_2 of a random variable x. Formally, the Kullback-Leibler measure is given by,

$$KL(p_1, p_2) = \sum_x p_1(x) \log \frac{p_1(x)}{p_2(x)}. \tag{5.4}$$

The Jensen-Shannon measure is then defined as,

$$JS(p_1, p_2) = \frac{1}{2} KL\left(p_1, \frac{p_1 + p_2}{2}\right) + \frac{1}{2} KL\left(p_2, \frac{p_1 + p_2}{2}\right). \tag{5.5}$$

$JS(p_1, p_2)$ is symmetric; i.e. $JS(p_1, p_2) = JS(p_2, p_1)$, however it is not a metric in the formal sense as it does not satisfy triangle inequality. In order to measure the distance between two spectral distributions we can take the square root of the distance,

$$D_{js}(f(\lambda), f(\mu)) = \sqrt{JS(f(\lambda), f(\mu))}, \tag{5.6}$$

where $f(\lambda)$ and $f(\mu)$ are two distributions on sets of eigenvalues λ and μ. D_{js} is strictly in the range $[0, 1]$ and satisfies the triangle inequality and is a metric in the formal sense [181]. However, given that $f(\lambda)$ is not a continuous distribution but either a histogram or a Gaussian kernel. This section uses the distance between 50 bins, the same number as used to generate the plots in Appendix Figure B.8.

Kolmogorov-Smirnov distance

Given that the cumulative distribution is a continuous space and does not require bins, we next define the distance between the eigenvalues of two normalised Laplacians as the Kolmogorov-Smirnov (KS) distance. This metric is used later in the chapter to measure the distance between degree distributions, a very different context not to be confused with the approach presented here. We define the cumulative eigenvalue distribution of a graph as

$$S_\lambda(x) = \frac{|\{\lambda_i | \lambda_i < x \wedge \lambda_i \in \lambda\}|}{|\lambda|}, \tag{5.7}$$

where x can take on any real value in the range $[0, 2]$. The KS distance between these two eigenvalue distributions is then defined as,

$$D_{ks}(\lambda, \mu) = \max_{x \in [0,1]} |S_\lambda(x) - S_\mu(x)| \tag{5.8}$$

As the KS distance only takes the maximal distance between two graphs, for many of the distributions shown in Appendix Figure B.9 this will likely be the central point $\lambda < 1$.

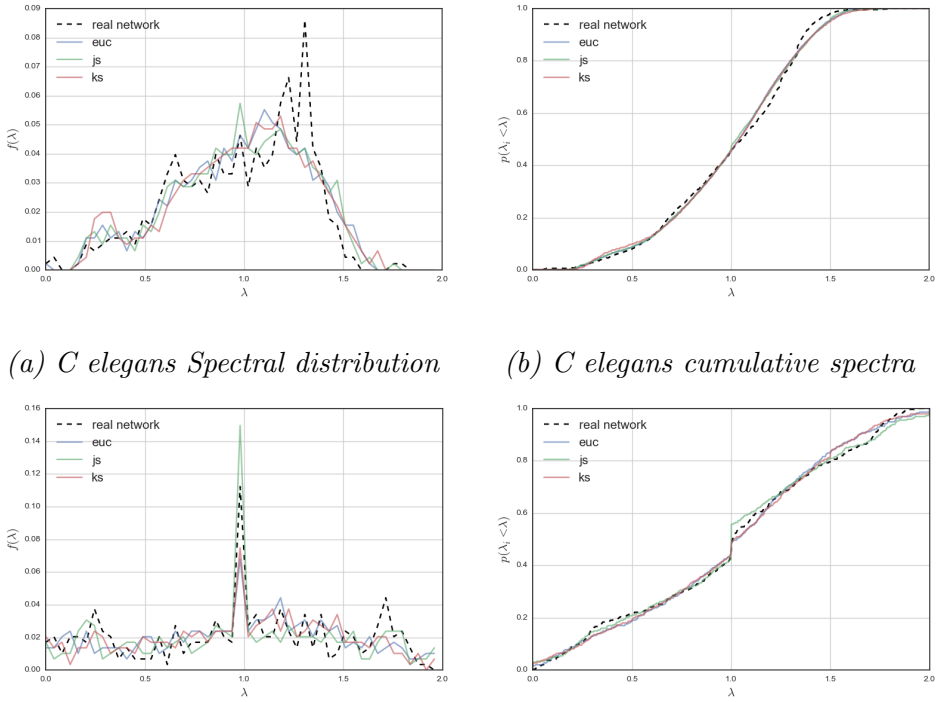
5.4.1 Fitting Metabolic Networks

In the following section, the feasibility of using the distance between spectral distributions of networks is evaluated using the *C elegans* and *E coli* metabolic networks. These networks are relatively small, but large and sparse enough to form a reasonable test of the procedure. Due to the time complexity of computing the eigenvalues of the normalised Laplacian, it was found that a fitting procedure was simply not feasible for the larger networks in this study. The PSO procedure in this process evaluates 30,000 different parameter sets, with a population size of 20. The objective of this process is to minimise each of the three distance metrics described in section 5.4.

As it was found that overlap appeared to have minimal impact on the graph spectra, it was fixed at 0 for these tests. This decision was made in order to reduce the dimensionality of the problem. The parameter boundaries for each network are determined as follows, each of the σ parameters is optimised in the range $[0.1, 2.2]$, K can take on any value between 1 and $n/10$, e_k is set in the range $[0, 0.5]$ and a is bounded between the values described in Table 5.1.

The results for the selected parameters, observed spectral distances and topological properties in the form of mean clustering assortativity and KS distance between degree distributions are shown in Table 5.2. The distances and topological measures are taken from 100 samples of the selected best fit parameters. Interestingly, each of the distance measures finds extremely different parameters for the best fit spectra. This indicates that CiGRAM is capable of generating similar graphs with very different parameters, this is likely due to the interaction between all of them. With this said, there do appear to be common patterns such as the high value of K in the *C elegans results* as well as high levels of e_k. Similarly, the a parameters tend towards the same values, with the exception of the Jensen-Shannon distance for *C elegans results* which places a on the upper boundary of the optimisation process.

Visually, example plots in Figure 5.1 indicate that the optimisation process was able to find close spectral matches for the resulting graphs. The results indicate that the JS distance was best able to match the peak in the *E coli* spectra. However, none of the distance metrics appear to be good representa-



(a) C elegans Spectral distribution

(b) C elegans cumulative spectra

(c) E coli Spectral distribution

(d) E coli cumulative spectra

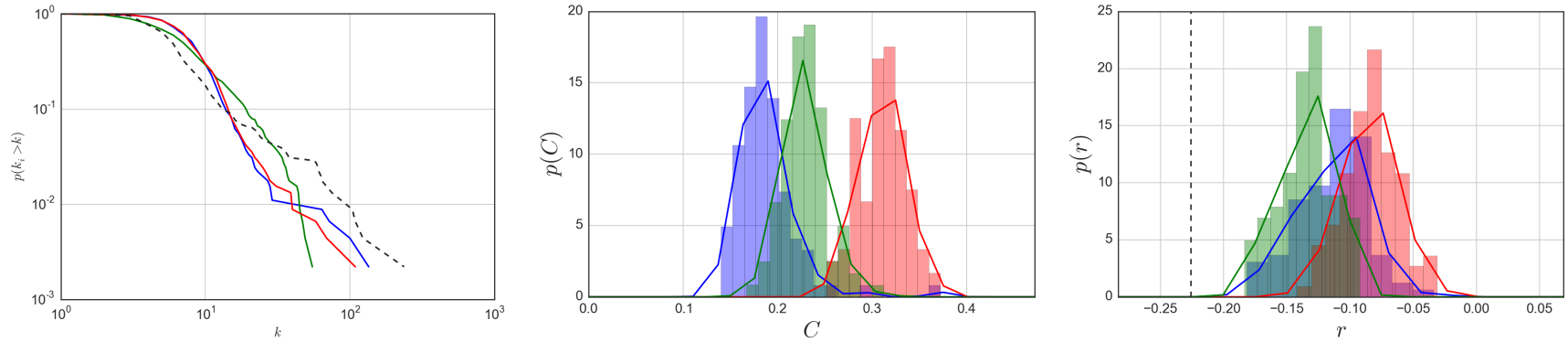
Figure 5.1: Example graph spectra of the best fit CiGRAM parameters for each distance metric.

tions of peak found in the *E coli*, indicating that CiGRAM may not be able to fully represent this graph.

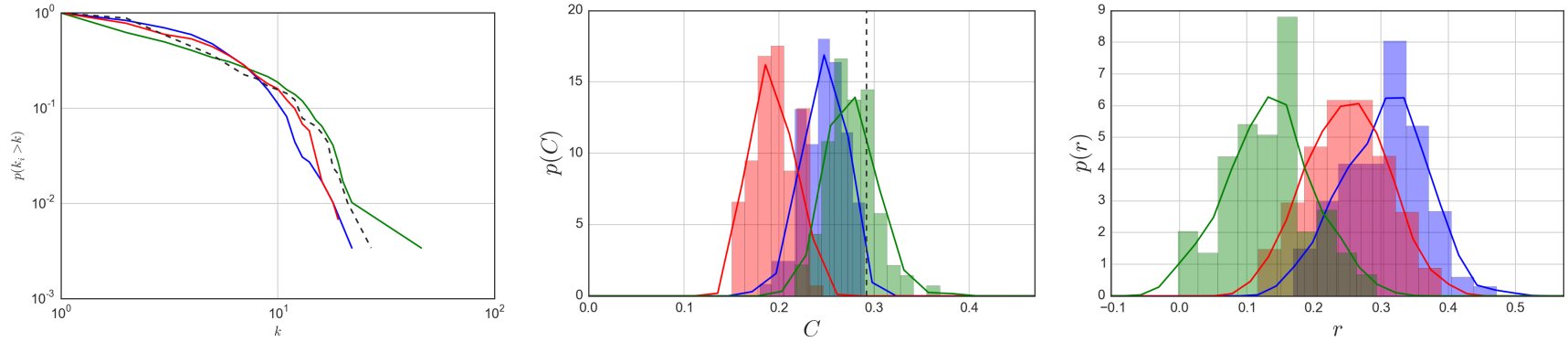
The resulting levels of assortativity, clustering and degree distributions appear to be very poor matches for the target graphs. The distributions of the results are shown in Figure 5.2, with kernel density estimates taken from the histograms. For *E coli* the level of clustering appears to be close to the target, shown in the dashed line. Whilst the degree distributions are visually similar they are, by no means, precise fits for the target networks. Assortativity does not appear to be well modelled by any of the target graphs.

5.4.2 Discussion of fitting network spectra

The spectral fitting evaluated in this section, at first, appears a promising approach to fitting networks. The fits here appear to be good spectral representations of the real world networks. However, several limitations must be noted. The spectral distribution is $O(n^3)$ complexity [192], making it unreasonable



(a) C elegans degree *(b) C elegans clustering* *(c) C elegans assortativity*



(d) E coli degree *(e) E coli clustering* *(f) E coli assortativity*

Figure 5.2: Topological properties of best spectral fits for metabolic networks generated with CiGRAM. The assortativity (r) and average clustering (C) results are taken from 100 replicates of the selected CiGRAM parameters.

Network	Metric	σ_f	$\tilde{\sigma}_s$	$\tilde{\sigma}_f$	$\tilde{\sigma}_s$	a	K	e_k	Dist	C	r	KS
E coli	euc	1.53	0.975	1.637	1.455	3.873	19.0	0.275	0.0	0.245	0.312	0.099
	js	2.2	0.716	1.214	1.121	3.219	28.0	0.478	0.135	0.194	0.251	0.222
	ks	1.673	0.883	1.75	1.195	1.773	23.0	0.252	0.056	0.267	0.142	0.127
C elegans	euc	0.491	2.2	0.553	0.946	-4.819	45.0	0.5	0.0	0.186	-0.112	0.281
	js	0.836	1.013	2.182	1.24	0.0	44.0	0.472	0.101	0.291	-0.093	0.153
	ks	1.135	1.923	0.765	1.199	-3.354	45.0	0.484	0.039	0.23	-0.138	0.267

Table 5.2: Fit of graph spectra for metabolic networks. The results for Distances (Dist), average clustering coefficient (C), degree assortativity coefficient (r) and Kolmogorov-Smirnov distance are means taken from 100 sample runs.

to use this approach on the larger datasets in Table 5.1. The nature of these distances is also that they are technically pseudo-distance metrics. In this respect, two networks could have a spectral distance of 0 yet have different topology. This is apparent with the results showing that, despite being a close match in terms of eigenvalues, the assortativity and degree distributions are a poor match. A further limitation is that it is hard to isolate and control the individual topological properties. For example, in Chapter 6 Section 6.4 the impact of assortativity on community detection algorithms is evaluated. Selecting a topological property to remain fixed whilst modifying other values is extremely difficult using the metrics described here.

5.5 Fitting summary statistics

The previous section evaluated the use of spectral distances as a form of cost function in the PSO optimisation of CiGRAM's parameters. The limitations of this method make it unsuitable for use in many of the large datasets that we would wish to analyse. The approach taken here is to use appropriate graph summary statistics that capture the behaviour of CiGRAM. For this purpose we use distance between the assortativity coefficient, clustering coefficient and the degree distributions. These summary statistics have been highlighted in the complex networks literature as vital aspects related to the structure of graphs [6, 7, 119]. However, they by no means capture all the topology of networks. This section is broken up as follows. The method of measuring the distance between degree distributions is presented in the form of the Kolmogorov-Smirnov distance and distance between maximum degrees. Then,

the formal summary statistic dissimilarity is computed as a combination of the degree, assortativity and clustering statistics.

5.5.1 Measuring degree distribution distance

The standard test for measuring the distance between the degree distributions is the *Kolmogorov-Smirnov* (KS) distance. The two sample KS tests take the maximal distance between two cumulative distributions with the objective of testing the null hypothesis that both are drawn from the same probability distribution. Here, we are interested in the test in order to measure the goodness of fit of a given model graph to the degree distribution of some observed target graph. Formally, the KS distance is given by,

$$D_{KS}(G, G') = \max_k |S_G(k) - S'_G(k)|, \tag{5.9}$$

where $S_G(k)$ and $S'_G(k)$ are the cumulative degree distributions of two networks and k indicates the node degree.

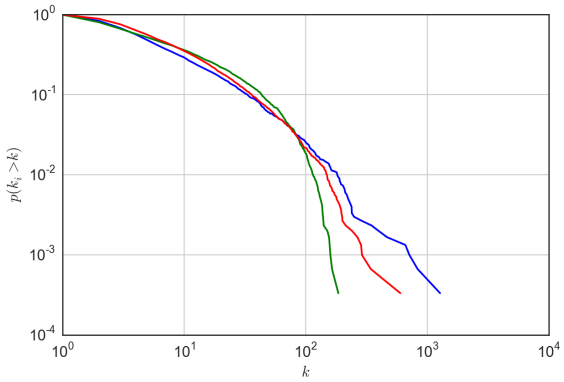


Figure 5.3: Complementary cumulative degree distributions highlights the insensitivity of the KS test to the extreme tails of distributions. Graphs have $n = 3000$*, a density of* 0.005 *and* $a = 0$*. Degree parameter vary at* $\sigma_f = 0.7, \sigma_s = 0.7$ *(blue),* $\sigma_f = 1.2, \sigma_s = 0.8$ *(green) and* $\sigma_f = 0.8, \sigma_s = 0.8$ *(red)*

For our purposes the Kolmogorov-Smirnov test seems appropriate, however, the distance is not uniformly sensitive across the range of the distributions. Most notably, the test is relatively insensitive to the distance between tails of the distributions [12]. The distance between two very different degree distributions is demonstrated in Figure 5.3, which shows three different complementary

cumulative degree distributions. 3 graphs of equivalent density generated by CiGRAM with different degree distribution parameters G are shown where $\sigma_f = 0.7, \sigma_s = 0.7$ (blue), H where $\sigma_f = 1.2, \sigma_s = 0.8$ (green) and I where $\sigma_f = 0.8, \sigma_s = 0.8$ (red). The KS distance between G and H is 0.08, whilst the KS distance between G and I is higher at 0.86. For G and H, the maximal degree is almost an order of magnitude different. This does not reflect the entire nature of the degree distribution, more important to many real world networks is the presence of a heavy right tail. The distances here do not reflect this, and initial tests showed that parameter optimisation of the KS test would often lead to extremely dissimilar networks. Furthermore, the general shape of I is closer to G and we would prefer an optimisation strategy to recognise this. Consequently, we introduction the log distance between the maximal degree of each graph,

$$D_{k_{max}}(G, G') = |\log_{10}(k_{max}(G)) - \log_{10}(k_{max}(G'))|, \tag{5.10}$$

where k_{max} indicates the maximal degree between two graphs. Equation 5.10 will only be sensitive to extreme changes in the maximal degree. Where the difference between the maximum degree of the two graphs is less than an order of magnitude $D_{k_{max}}$ is necessarily less than 1. The objective for the optimisation procedure is to find the general trend of the degree distribution, rather than getting trapped in local minima that poorly represent the whole graph structure.

Computing degree distribution fit quality

As the ultimate objective of the following sections is to maximise the similarity, we design the following hypothesis test to see if a given degree distribution is adequately described by a given model. This approach is similar to the one taken by Clauset *et al.* [12] to see if a given power law model fits a dataset. In this study, however, we lack the ability to generate a high number of permutations given that computing a large number of graphs with CiGRAM is computationally infeasible. The expected cumulative distribution of a model can be calculated from a suitably sized sample of CiGRAM with a given parameter set. The average cumulative distribution function (ACDF) is determined as

the expected value for each value of $S_G(k)$ from a suitably large number of degree distributions generated by the model. The sample size of 100 degree distributions was found to be large enough to create a good level of accuracy, requiring more than 30 samples to form a representative distribution. The trade off is deciding between a more precise ACDF and a sample set that is computable given the size of the networks. The KS distances of each of the 100 samples distributions to the model ACDF are then computed, giving a representative distribution of model error.

The variance of the model error from the ACDF appears to be gaussian, allowing the use of a standard z-test for the distance of the target distribution from the ACDF of CiGRAMs model parameters. The null hypothesis is that the target degree distribution is within the margin of error for the model. In this situation, the alternative hypothesis is that the model does not represent the observed degree distribution. In order for the model to be considered to generate appropriate degree distributions, we must not reject the null hypothesis at $p > 0.1$.

5.5.2 Summary statistic distance

Given the description for the distance between degree distributions above, we can now define a formal dissimilarity measure between two graphs. Since the objective of CiGRAM is to form a representative model of the community structure and assortativity of a graph, as well as the degree distribution, we also use the degree assortativity coefficient and clustering coefficient in the dissimilarity measure. The graph dissimilarity measure is defined as,

$$f(G, G') = 2D_{KS}(G, G') + D_{k_{max}}(G, G') + |r_G - r_{G'}| + |C_G - C_{G'}|, \quad (5.11)$$

where $r_G, r_{G'}$ and $C_G, C_{G'}$ are the degree assortativity (see Equation 2.26) and average clustering coefficients (see Equation 2.4) of the two graphs G and G', respectively. In principle, the KS test can take on a value as large as 1. However, in practice, it was found that for the heterogeneous configurations of interest, the KS distance was less than 0.2, for this reason the measure is doubled compared with the other summary statistics.

One key issue with the above fitness measure is the computation time of

the topological properties, particularly the average clustering coefficient of the network C. Computing the clustering coefficient of an individual node, C_i, requires cycling through all secondary neighbourhoods of a node to check for reciprocation, the computation is therefore $O(n^2)$ in complexity [193]. To improve upon this we use the sampling approach of Schank and Wagner [193] which gives an estimate of C, runs in $O(1)$ time and has been shown to have good levels of accuracy. This algorithm samples 1000 nodes, with replacement, from V and then samples 2 neighbours from each of the sampled nodes. Transitive closures are then recorded as triangles. The estimated mean clustering coefficient is then the number of observed triangles divided by the number of samples.

Under certain test conditions there is no interest in matching the observed community structure of a given graph, for this reason we include a second dissimilarity fitness measure,

$$f_c(G, G') = 2D_{KS}(G, G') + D_{k_{max}}(G, G') + |r_G - r_{G'}|. \tag{5.12}$$

We consider Equation 5.12 as the *null community* fitness, in order for a graph to be determined to have significant community structure it must have a significantly higher clustering coefficient than the best fit CiGRAM where $K = 1$.

5.5.3 Graph parameter tests

The following subsections define a number of approaches to fitting the empirical networks described in section 5.3. These approaches all have the same objective in mind; to fit the summary statistic cost functions described in equations 5.11 and 5.12. Evaluating the ability for the model to describe the topological summary statistics follows in Section 5.5.4. For every desired graph summary statistic (degree distribution, assortativity and average clustering) we are testing the null hypothesis that the observed empirical value is described by the target model. In this situation, the alternative hypothesis is that the model parameters do not describe the topological summary statistic observed in the empirical graph. The reader should be aware of the implications of this form of model validation in that no model can ever be said to perfectly describe the data under these test conditions. For any given selected parameter set,

it is always possible to argue that a better fit set of parameters can exist or another hypothetical model or models exist that better describe the empirical data. For each of these test sets, the PSO optimisation is completed with 8000 evaluations. It was found, when completing the spectral fit procedure, that this was sufficient to guarantee convergence for the fit (results not presented). The best fit parameters are selected with 100 additional model replicates from the 20 best observed parameter sets, this ensures that the initial 5 evaluations in the PSO process are not just "lucky" and selects the best overall performing candidate.

In this approach, for the clustering coefficient and assortativity coefficient, the null hypothesis is tested under the condition that the observed topological property in question must fall within two standard deviations of the best fit model. In Section 5.5.4, for target assortativity, maximum degree and average clustering this is phrased in terms of a z-test; if the null hypothesis is rejected with probability $p < 0.05$. Given that the true value of the variance is not known for CiGRAM, for model validation purposes we are stricter than this and say require $p > 0.1$ to consider the summary statistic to be adequately described by the model. The p-value for the KS distance is described in Section 5.5.1. The issue for model fit, from this perspective, is that the values for the network are only a single value and not drawn from a distribution. Consequently, the p-value only represents the probability that the empirical value would have been generated by the model.

Single K

The simplest conditions for fitting networks explored in this chapter is to avoid optimising the community structure of the network and to only attempt to fit the degree distribution and assortativity of the empirical network. For this approach, the fitness function used does not include the clustering coefficient (see Equation 5.12). This experiment condition provides a formal hypothesis test for the presence of block structure. Assuming other topology is well represented, under these test conditions if the clustering coefficient rejects the null hypothesis we can argue that block structure is required to generate the target network. In other words, if the average clustering coefficient of a

graph generated with $K = 1$ is not as high as that of the empirical graph, the empirical graph can be said to have some form of block structure and can be divided into modules. Alternatively, if the clustering coefficient of the $K = 1$ class of model is as high as the empirical graph there is no evidence of modular structure in the real world network.

Fixed K and e_k

This experiment condition tests to see if simply fitting the block structure and fraction of edges between communities is adequate to generate the observed clustering of real networks. In Chapter 4 it was shown that K has a direct influence on the clustering coefficient networks generated on CiGRAM. The observed clustering coefficient is influenced by parameters that influence the size and density of other communities. This, however, may not be relevant to the level of clustering in the graph and it may be possible to generate the observed clustering simply by specifying the correct number of blocks. This would demonstrate that the average clustering summary statistic is insensitive to the distribution and size of the blocks in question.

It is difficult to *a priori* know the value of K to select. This leaves two alternatives, to estimate K and e_k with other parameters, or to use some heuristic to estimate the values of k and e_k in the real graph. For this test we opt to use the results of multiple runs of the Louvain community detection algorithm [82] to generate a range of partitions.

The Louvain algorithm greedily agglomerates nodes that increase the modularity score starting from a random partition. Each random starting partition, therefore, results in one of many local optima. In order to sample the local optima space one only needs to maximise modularity starting from a random partition. We generated 100 random starting partitions by sampling from the random set of cut sets and take the unique, resulting locally optimum partitions. Given the local optima partitions generated by the Louvain algorithm, we take K to be the median number of detected modules and e_k to be the mean fraction of edges partitioned between communities observed.

The objective of using fixed K and e_k is to see if the performance of the optimiser can be improved with fixed parameters. As the clustering coefficient

cannot be directly influenced in the model, the fitness function used is Equation 5.12, the null community fitness.

Free parameters

The objective of the free parameter approach is to fit CiGRAM as well as possible with respect to the graph similarity metric. In the free parameter approach, the parameters of CiGRAM are allowed to take on any value, within the bounds of the experiment. The objective here is to provide an adequate test model ensemble for community structure evaluation in Chapter 6. In order for the model to be valid it must not reject the null hypothesis for each of the summary statistics used to fit the model. This experiment should be seen as an overall test of the ability of CiGRAM to represent topological properties of target networks. In this context, CiGRAM is only a sufficient fit for a given topological summary statistic if the null hypothesis cannot be rejected.

Fixed e_k and p_o

In order to benchmark community detection algorithms, a topic described in detail in Chapter 6, it is desirable to be able to create noisy communities that have configurable levels of mixing between communities. The objective is to find the best fit for topological properties with a fixed level of e_k and p_o. These settings greatly restrict other topological properties of the graph. This forms two types of experiments: low overlap with $e_k = 0.1$ and $p_o = 0.05$ and high overlap with $e_k = 0.25$ and $p_o = 0.1$. The null hypothesis for fitting under these conditions is that the CiGRAM model is unable to find best fit parameters for real world graphs.

5.5.4 Assessing fit quality

The following section describes the fit quality of CiGRAM under the varying parameter conditions described in the previous section. The results for the optimisation process are shown in Table 5.3 with distribution of average clustering coefficients in Figure 5.6 and degree assortativity coefficients shown in Figure 5.5. These results are taken from 100 replicates of the best fit parameters,

giving an indication of the model fit. In assessing model fit we are limited in the approach taken here given that the observed values and distributions are based only on a single sample of each topological summary statistic. No judgements can be made about the underlying processes that make up the resulting graphs, and therefore the true distribution that these statistics are drawn from. The p-values in Table 5.3 are used to make judgements about model fit, for assortativity and clustering these are simply based on the z statistic. As stated previously, a value of $p > 0.1$ indicates that the null hypothesis cannot be rejected, meaning the model is a plausible fit for the real world topology.

The parameters that give rise to these values are found in Appendix Table B.1. The lack of similarity between the different parameter sets across experiments indicates that the search space may have many locally optimal solutions. A judgement that a given model is the best possible fit can only be made if the search space is exhausted such that all parameter sets have been sampled. Consequently, the results presented here can only be interpreted as one of many potential plausible models.

Single K

The results for the single K experiment indicate that every network has a statistically significant level of clustering with respect to the null community free model of CiGRAM. This assessment can be made given that the degree distribution and assortativity do not reject the null hypothesis of being plausible fits for the network. This indicates that all the networks have an underlying structure that increases the dependence between subsets of vertices leading to transitive closures. This follows on from the assumption in the previous Chapter that a community is equivalent to a random graph and that increased dependency between vertices is required for a large number of triangles. However, not all the networks have an appropriately fitting degree distribution. For example, the Hamster network, SeedNet and PGP reject the null hypothesis that the degree distribution generated by the best fit parameters could have given rise to the empirical degree distribution. This result is interesting considering that other experiments with community structure appear to reject the alternative hypothesis for other networks. It may be that CiGRAM has difficulty fitting

Experiment	Network	$\hat{C}$	p	$\hat{r}$	p	$\hat{k_{max}}$	p	$\hat{D_{KS}}$	p
Single K	Yeast PPI	0.011	0.0	-0.097	0.553	64.5	1.0	0.016	0.677
	Arabidopsis PPI	0.012	0.0	-0.222	0.277	209.9	1.0	0.013	0.555
	C Elegans Metabolic	0.156	0.0	-0.273	0.142	259.2	1.0	0.022	0.929
	E coli Metabolic	0.067	0.0	0.611	0.534	27.03	1.0	0.03	0.745
	SeedNet	0.145	0.0	0.177	0.492	1565.35	1.0	0.056	0.0
	Open Flights	0.066	0.0	0.047	0.465	255.0	1.0	0.019	0.147
	US Power Grid	0.0	0.0	0.01	0.706	20.8	1.0	0.001	0.932
	PGP	0.002	0.0	0.248	0.661	206.76	1.0	0.022	0.0
	Hamster	0.091	0.0	-0.096	0.341	265.08	1.0	0.027	0.03
Fixed K	Yeast PPI	0.102	0.0	-0.102	0.428	63.48	1.0	0.017	0.666
	Arabidopsis PPI	0.029	0.0	-0.198	0.504	293.78	1.0	0.011	0.757
	C Elegans Metabolic	0.107	0.0	-0.264	0.182	256.84	1.0	0.033	0.705
	E coli Metabolic	0.081	0.0	0.509	0.361	33.95	1.0	0.028	0.841
	SeedNet	0.285	0.001	0.154	0.353	1450.27	1.0	0.036	0.726
	Open Flights	0.118	0.0	0.026	0.305	256.1	1.0	0.025	0.103
	US Power Grid	0.013	0.0	0.023	0.925	18.53	1.0	0.002	0.965
	PGP	0.088	0.0	0.235	0.465	200.81	1.0	0.006	0.504
	Hamster	0.262	1.0	-0.122	0.059	275.8	1.0	0.02	0.646
Free params	Yeast PPI	0.139	0.693	-0.08	0.863	63.51	1.0	0.013	0.836
	Arabidopsis PPI	0.131	1.0	-0.141	0.965	354.2	1.0	0.009	0.609
	C Elegans Metabolic	0.538	0.141	-0.27	0.168	169.75	1.0	0.068	0.322
	E coli Metabolic	0.286	0.48	0.516	0.225	26.19	1.0	0.03	0.952
	SeedNet	0.454	0.306	0.048	0.12	1192.42	1.0	0.067	0.124
	Open Flights	0.442	0.197	0.02	0.0	256.37	1.0	0.044	0.0
	US Power Grid	0.08	0.508	0.009	0.602	19.21	1.0	0.001	0.983
	PGP	0.248	0.154	0.236	0.507	190.02	1.0	0.005	0.842
	Hamster	0.129	0.316	-0.091	0.466	362.43	1.0	0.039	0.002
Low overlap	Yeast PPI	0.135	0.525	-0.094	0.616	57.11	1.0	0.016	0.801
	Arabidopsis PPI	0.337	1.0	-0.207	0.427	235.26	1.0	0.014	0.63
	C Elegans Metabolic	0.548	0.047	-0.343	0.017	202.45	1.0	0.029	0.621
	E coli Metabolic	0.206	0.0	0.505	0.142	24.85	1.0	0.028	0.58
	SeedNet	0.341	0.056	0.087	0.169	1309.66	1.0	0.044	0.538
	Open Flights	0.476	0.797	0.048	0.472	359.88	1.0	0.057	0.82
	US Power Grid	0.087	0.933	-0.018	0.096	17.41	0.998	0.002	0.987
	PGP	0.264	0.457	0.214	0.187	206.74	1.0	0.008	0.646
	Hamster	0.107	0.166	-0.075	0.76	312.01	1.0	0.026	0.143
High Overlap	Yeast PPI	0.134	0.485	-0.095	0.587	61.16	1.0	0.019	0.39
	Arabidopsis PPI	0.127	0.998	-0.225	0.322	256.09	1.0	0.015	0.211
	C Elegans Metabolic	0.47	0.0	-0.296	0.098	162.67	1.0	0.056	0.0
	E coli Metabolic	0.06	0.0	0.626	0.654	26.37	0.999	0.029	0.852
	SeedNet	0.354	0.034	-0.028	0.0	1169.42	1.0	0.041	0.47
	Open Flights	0.439	0.362	0.032	0.328	267.1	1.0	0.015	0.415
	US Power Grid	0.076	0.168	-0.016	0.098	19.15	1.0	0.002	0.96
	PGP	0.251	0.03	0.071	0.0	191.08	1.0	0.014	0.249
	Hamster	0.159	0.962	-0.11	0.061	229.05	1.0	0.02	0.223

Table 5.3: Best fit CiGRAM results. Results show expected mean clustering coefficient $\hat{C}$, expected degree assortativity $\hat{r}$, expected maximum degree $\hat{k}_{max}$ and mean Kolmogorov-Smirnov distance D_{KS}. Results are taken from 100 samples of CiGRAM with the best fit parameters in Table B.1. p-values for $\hat{C}$, $\hat{r}$ and $\hat{k}_{max}$ are taken from a two sided z-test, where $p > 0.1$ we cannot reject the null hypothesis that the value was drawn from the model.

these degree distributions due to the nature of the highly clustered, latent modular structure.

Fixed K

Interestingly, the results for the fixed K and e_k tests indicate that simply defining the block structure is not sufficient to generate the average clustering found in real world networks. For all networks but the Hamster graph, the block structure is not well represented simply by ensuring that the degree connectivity matches that of a real world graph. This indicates that both the correct level of K and the specific distribution sizes of the communities are required to be fitted for a block structure to be accurate. In the case of the degree corrected stochastic block model [130], the internal structure of the groups is not considered a property of the model. Indeed, conventional block modelling assumes that the internal communities are largely homogeneous in size.

Free parameters

The results for the *free parameters* fitting set indicate that CiGRAM is capable of fitting the desired topological properties of the graphs studied here. Under these conditions all the tests have $p > 0.1$ indicating that the null hypothesis cannot be rejected. This does not imply that CiGRAM is a perfect model for these graphs. Indeed, later in the Chapter we demonstrate that the approach of using these summary statistics is insufficient to fit all topological properties. However, the objective of fitting every imaginable topological feature is both difficult to model and would certainly require an advanced distance metric. The objective of CiGRAM is to form a model capable of allowing domain specific module detection algorithm selection, these results appear to match this goal. The Open Flights network is the only graph for which the resulting model distributions are significantly different from the observed empirical degree distribution and assortativity. Whilst the fit here is poor, the model is still a relatively close representative of the real networks topological properties. A KS distance of 0.044 is relatively small in this context and the degree assortativity of 0.02 is relatively similar to the target of 0.049. In the following Chapter the

results for community detection analysis need to be considered in the context that even an evaluation under an inaccurate model is better than no model at all.

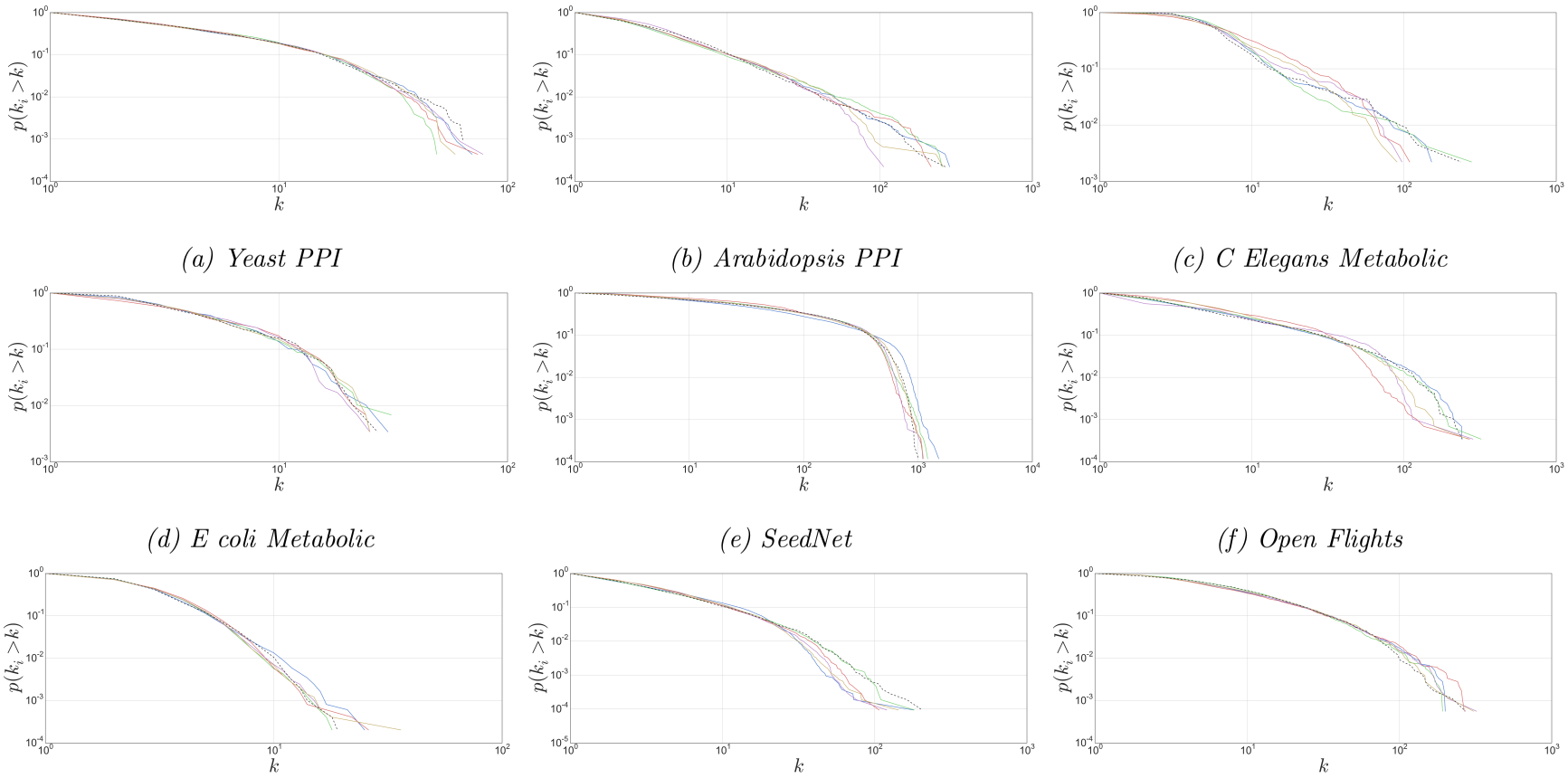
Fixed overlap

When fixing a specific level of overlap, the ability of the fitting procedure to match all relevant topological properties is more challenging. p_o and e_k were previously shown to place strong bounds on the community structure of networks. This, likely, explains the worse clustering coefficients found in the high overlap results. However, when comparing the results with the clustering coefficients from the single K results they are significantly higher. Given that the other topological properties are still reasonable representations of the empirical graphs, these model parameters still have value when considering community detection algorithms. The level of overlap between communities should be considered to be noisy as this allows one to evaluate the performance of algorithms in less than ideal situations.

Fitting other topological properties

The best fit results show that CiGRAM can fit the degree distributions, assortativity and clustering coefficients of real world graphs through a geometric model including community structure. However, these topological properties are explicitly included in the objective function optimised by the PSO process. Included in Appendix B are results relating to additional topological properties not explicitly modelled under the conditions of CiGRAM. The objective here is to test if the summary statistics, alone, are capable of generating richer topology. Measured are the mean shortest path length (see Equation 2.3), central point dominance (see Equation 2.6) and modularity (see Equation 2.11). Results are shown in Appendix Table B.2 and Appendix Figures B.5 to B.7, with p values calculated in the same manner as the assortativity and clustering coefficients in Table 5.3.

The mean shortest path length of the networks is a crucial topological property, given that it determines the ability for messages to pass between nodes. For SeedNet the results are the furthest from those generated under

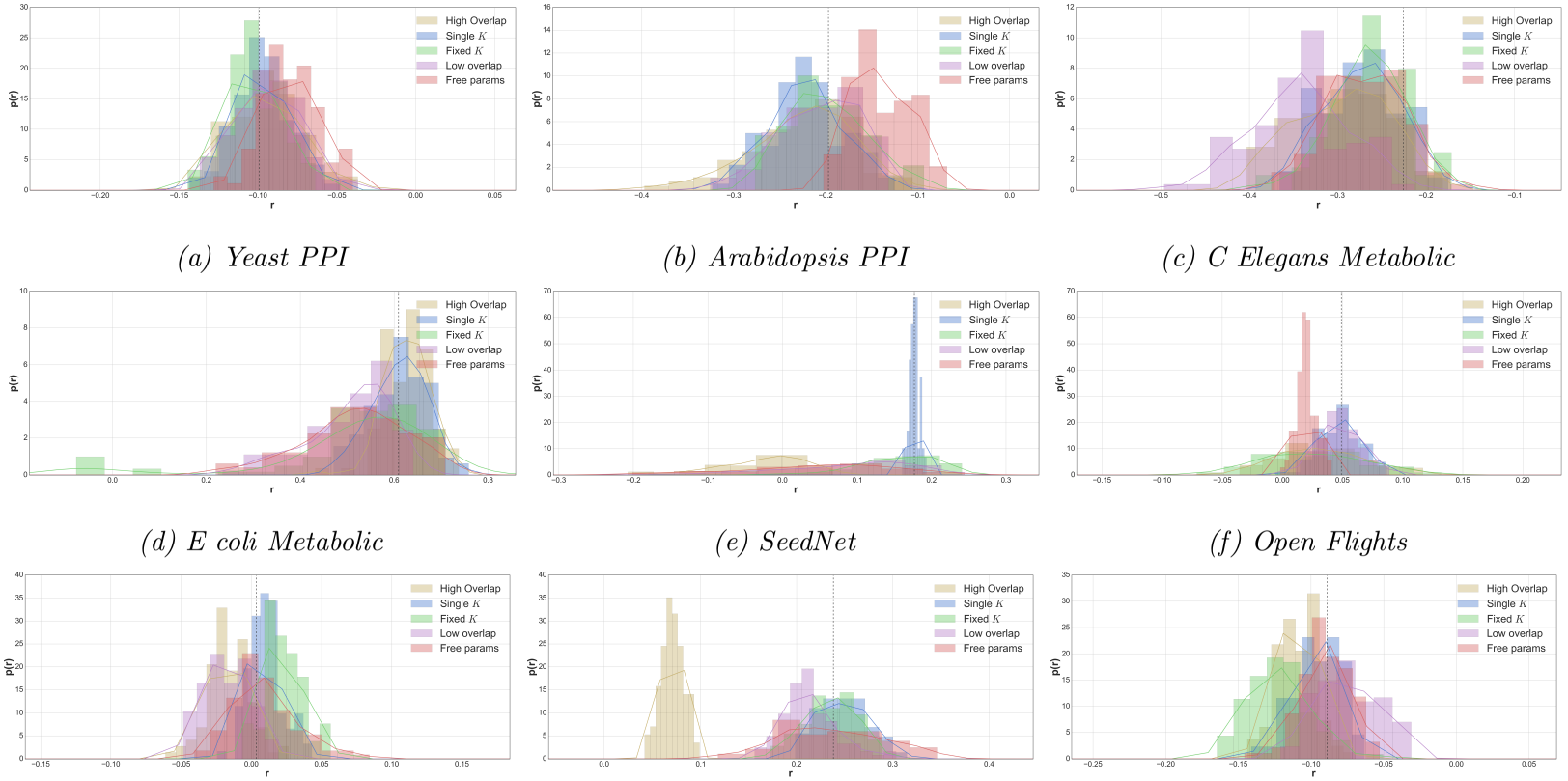


(a) Yeast PPI *(b) Arabidopsis PPI* *(c) C Elegans Metabolic*

(d) E coli Metabolic *(e) SeedNet* *(f) Open Flights*

(g) US Power Grid *(h) PGP* *(i) Hamster*

Figure 5.4: Degree distributions for best fit models and compared to real world graphs. Colours indicate model fit for real world graphs (black dashes), single K (blue), fixed K (green), free parameters (red), low overlap (purple) and high overlap (yellow).

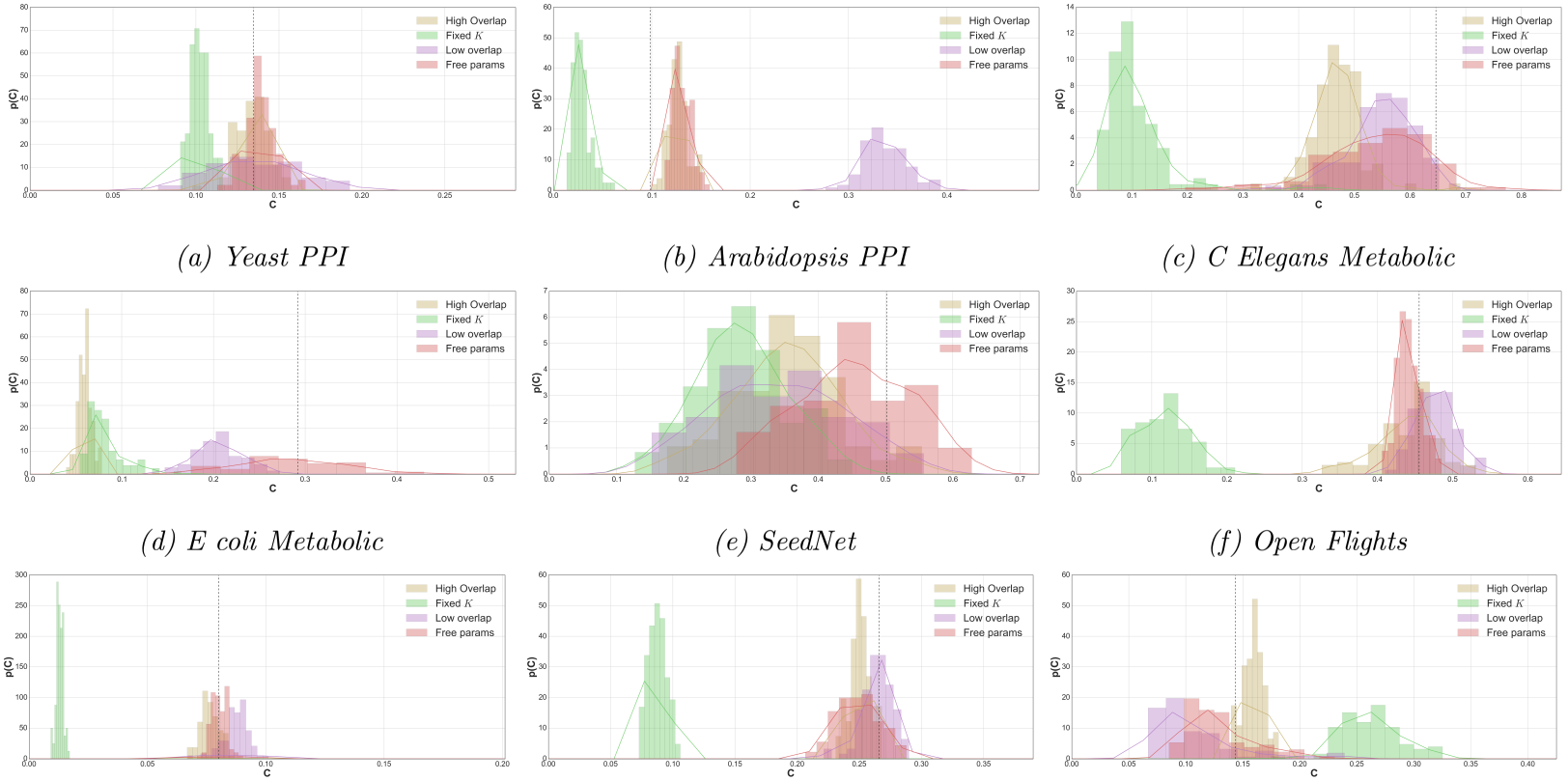


(a) Yeast PPI *(b) Arabidopsis PPI* *(c) C Elegans Metabolic*

(d) E coli Metabolic *(e) SeedNet* *(f) Open Flights*

(g) US Power Grid *(h) PGP* *(i) Hamster*

Figure 5.5: Distribution of degree assortativity coefficient for best fit models, histograms of 100 samples with kernel density estimates are shown. Dashed black lines indicate values observed in real networks. Colours indicate model fit for real world graphs (black dashes), single K (blue), fixed K (green), free parameters (red), low overlap (purple) and high overlap (yellow).



(a) Yeast PPI *(b) Arabidopsis PPI* *(c) C Elegans Metabolic*

(d) E coli Metabolic *(e) SeedNet* *(f) Open Flights*

(g) US Power Grid *(h) PGP* *(i) Hamster*

Figure 5.6: Distribution of mean clustering for best fit models histograms of 100 samples with kernel density estimates are shown. Dashed lines indicate values observed in real networks. Colours indicate model fit for real world graphs (black dashes), single K (blue), fixed K (green), free parameters (red), low overlap (purple) and high overlap (yellow).

CiGRAM with a similar scale for the US Power Grid. Interestingly, the result generated by CiGRAM is significantly lower than the real network for both cases. Both of these networks have an interesting geometric interpretation, the US Power Grid being constrained by the geographic positions of the vertices and SeedNet by the geometric nature of correlation networks discussed in Chapter 3. These results may indicate that in order to more efficiently model shortest paths of networks of this form, a geometric condition should be applied. In the other networks, the shortest path length cannot be said to be accurately modelled simply through the optimisation functions. Though it is known that the level of clustering and the degree distribution has an impact on the shortest paths [5], these properties, alone, are not sufficient to accurately model the shortest paths of all networks.

CPD is a measure for the dependency of a network on one or two nodes with high levels of betweenness centrality (see Equation 2.5). Naturally, this is strongly related to the mean shortest path length. Unsurprisingly, the CPD of SeedNet and the US Power Grid is significantly lower than any of the models. This is likely due to the long shortest path lengths requiring a small number of nodes that have high betweenness centrality. For other networks, CPD is modelled more accurately though, notably, the PGP network has a significantly lower dependence on critical nodes than found in the modular CiGRAM models, yet significantly higher than the $K = 1$ class of models. This indicates that the modular structure has an impact on this score but that it is not accurately modelled through the approach taken in this chapter.

Treating modularity as a measure, rather than an approach to uncover any block structure, gives insight into the block structure generated by CiGRAM. The results here, again, show that the summary statistic based fitting approach is insufficient to fit the observed real world networks. One may expect the Fixed K experiment to generate appropriate results given that e_k and K both strongly relate to the level of modularity in a real network. However, this does not appear to be the case, and the results show that modularity is no better represented for the Fixed K models than any other parameter sets.

Additionally, results are included for the spectral distances used in section 5.4. Plots of the normalised Laplacian spectra are shown in Appendix Figure

B.9 and B.8, with the distances included in Table B.2. The graph spectra appear to be poorly represented by this fitting approach. A direct comparison to between the metabolic networks in Figures 5.1 and B.8 highlights that the summary statistics method does not appear to be a good approach to fitting network spectra.

5.5.5 Results summary

The results for the Single K experiments show that all the networks appear to have statistically significant clustering that cannot be explained by non-modular random graphs. The assortativity is fit for all models and, with the exception of SeedNet, PGP and Hamster, the degree distributions are very close when generated with fixed $K = 1$ model conditions. The fixed K results show that these parameters are not sufficient to generate the desired clustering coefficients and K and e_k, alone, do not allow accurate representations of real world data. The free parameters model highlights that CiGRAM is capable of fitting the desired topological properties for real world networks. Fixing the level of overlap and the fraction edges between communities makes fitting all the topology a challenge as these parameters place strong bounds upon the community structure that is generated. In terms of other topological properties, the fitness functions used in this work highlight that it is necessary but not sufficient to fit other topology such as mean shortest path length, centrality and modularity.

5.6 Chapter summary

In this chapter we have developed a fitting methodology for CiGRAM that uses particle swarm optimisation combined with a graph dissimilarity measure in order to fit key topological properties of degree correlations, degree distributions and mean clustering coefficients. The fit of these networks demonstrates that CiGRAM is capable of generating these salient features. However, this analysis uncovered several limitations. Namely, the fitness function used is unable to capture other topological properties of graphs such as the mean shortest path lengths and central point dominance. The Kolmogorov-Smirnov distance has

clearly demonstrable insensitivity to the tails of distributions which makes it very limited in the case of heavy tailed distributions used here. The fits to real world networks still appear to be relatively good in the plots shown in Figure 5.4.

Given the inexact nature of the fitness function, it is difficult to assess the success of the PSO algorithm. The results demonstrate that it is capable of dealing with the high dimensional noisy domain for which the gradient of the fitness function is unknown. However, many of the selected parameters appear to be on the edge of the accepted values; this may indicate that a wider range of parameters is required for networks. Parameters with lower levels of variance for the summary statistics used in the dissimilarity measure, such as more uniform community sizes, appear to cause the PSO algorithm to get trapped in local optima.

The following Chapter uses the fitting approach described here as a form of test bed for benchmarking graphs with community structure. Given that the community structure of CiGRAM is known, the best fit graph allows the analysis of algorithms in domain specific contexts.

Chapter 6

Benchmarking module detection algorithms

6.1 Introduction

In Chapter 3 we observed a lack of agreement relating to a number of community detection algorithms when applied to large scale plant correlation of expression networks, which indicates algorithm selection is a critical problem. In the case of social networks, ground-truth meta-data for real modules exists and can be used to validate algorithms [194]. However, meta-data for biological networks is extremely variable and, as a consequence, validation and selection of algorithms is an extremely challenging task. Moreover, the importance of valid module detection algorithms in biological networks should not be underestimated. Many biological networks are used for the generation of new hypotheses for functionally related genes [44] or protein complexes [4]. Therefore, methods are required to validate and improve the selection of algorithms.

This chapter aims to answer two core research questions:

- Does assortativity impact the performance of community detection algorithms?
- For a given network, which module detection algorithm is the best choice?

The main aim of this thesis is to provide a mechanism for evaluating the performance of module extraction algorithms in a domain specific context. In

this Chapter we consider the CiGRAM model, presented in Chapter 4, as a benchmark graph, given a known real community structure that can be considered a *"ground-truth"* set of communities. In order to create benchmark graphs, however, all modules must be internally connected subgraphs. This is achieved through a simple rewiring procedure that is guaranteed to produce connected graphs.

Assuming that that the reader is now more familiar with the CiGRAM modelling approach covered in Chapter 4, this chapter begins with a brief comparison to the Lancichinetti-Fortunato-Radicchi (LFR) benchmark models [9,128], discussed initially in Chapter 2 Section 2.5.5. This provides some insight into the current gold standard approach currently used to evaluate community detection algorithms. The limitation of the LFR benchmark, however, is that it is not designed to be a representation of real world graphs and so cannot be used directly for validation purposes. The work of Orman et al. [195] provides further insight into the importance of realistic models for community detection approaches. By making changes to the LFR benchmark, the authors of [195] were able to include network structure from alternative models that more closely resembled the topology of networks found in empirical data. These models, however, are still limited in their ability to represent the properties of real world networks when compared to the fitting approaches described in Chapter 5.

The chapter then focuses on degree assortativity, a property not modelled by the LFR benchmarks. Under controlled conditions (making use of the optimisation from Chapter 5), this topological feature is found to impact certain algorithms. This indicates that networks with degree-degree correlations, like those studied in Chapter 3, may be inappropriate for some of the approaches tested here.

The final section of this Chapter discusses a new methodology for module detection algorithm selection in the context of best fit models. By using appropriately fit models from Chapter 5, the selection of algorithms can be made in a more informed manner than simply using generic benchmarks. A core discovery of this section is that, as with recent results on the performance of ground-truth social and information networks [194], the algorithms evaluated

here fail to perform well in this more realistic context.

6.2 Comparison with the LFR benchmark

In the following section we describe the LFR benchmark graph in contrast to CiGRAM. The LFR benchmark is based on a planted partition model, in which nodes are given a fixed target degree and a community. The network is then wired, under the conditions of the configuration model (described in Section 2.5.3), with the added constraint that a certain percentage of a node's adjacent edges must be inside a pre-assigned grouping.

The LFR benchmark has a number of parameters that determine the topology of the generated graph. The parameters are outlined in Table 6.1. The γ parameter determines the power law exponent used to generate fixed degree distributions; this is analogous to σ_f and σ_s in CiGRAM. The κ parameter determines the power law exponent used to generate the community sizes, modelled by $\tilde{\sigma}_f$ and $\tilde{\sigma}_s$ in CiGRAM. Notably, these parameters force the generation of power law degree distributions. As previously noted in Chapters 3 and 5, many of the real world biological and non-biological networks are not best described by power law degree distributions. Assuming that networks are scale-free appears to be too strong an assumption for the selecting community detection algorithms. Even in the cases where the degree distribution does appear scale-free, this is only an approximation for the tail of the distribution.

In terms of the mixing between communities, determined in CiGRAM by e_k (the fraction of edges between communities) and p_o (the probability of overlapping nodes), LFR has comparable parameters. μ is determined by the mixing coefficient which determines the percentage of edges that each vertex will have between communities. That is to say, $1 - \mu$ fraction of neighbours for each node will exist between communities. This is a slight variation on the approach CiGRAM takes, as the number of edges between communities is not fixed and will vary depending on other parameters. As μ is based on individual nodes, a value of $\mu > 0.5$ implies that a node is more likely to connect with edges between communities than within its own community. In this sense, it is difficult to argue that a graph generated with $\mu > 0.5$ has a

Symbol	CiGRAM Analogue	Description
γ	σ_f and σ_s	Degree distribution power law exponent
κ	$\tilde{\sigma}_f$ and $\tilde{\sigma}_s$	Community size distribution
μ	e_k	Mixing parameter
o_n	p_o	Number of overlapping nodes
o_m		Number of overlapping memberships
c_{min}	c_{min}	Minimum community size
c_{max}	None	Maximum community size

Table 6.1: Parameters of the LFR benchmark

strict community structure. Indeed the authors of [9, 128] recognise this and state 0.5 as a threshold.

A further distinction between CiGRAM and LFR is that the selection of communities is determined by node degree; a node cannot be a member of a community if the average internal degree in that community is significantly different from its own. This has a strong implication that is also adopted by the degree corrected stochastic block model [130]. In CiGRAM, node position is determined independently from the communities, and the resulting degree emerges as a product of the assignment of community size and density.

The overlapping parameters o_n and o_m determine the number of nodes that exist in more than a single community and the number of communities they exist in, respectively. These parameters are very different in form to the p_o parameter as they are far more controlled than CiGRAM's analogue.

The lack of a fixed parameter with regards to the number of communities K is a further difference between the LFR benchmark and CiGRAM. Whilst not tested in this study, in certain cases, there may be a call to use and test community detection algorithms that use a fixed number of clusters as a parameter such as fuzzy-c-means [78].

The use of fixed degree distributions is a significant advantage over CiGRAM, which has to use a fitting procedure. Though the actual model tested here uses fixed power law distributions, allowing it to fit any degree distribution would be a trivial change to the LFR benchmark.

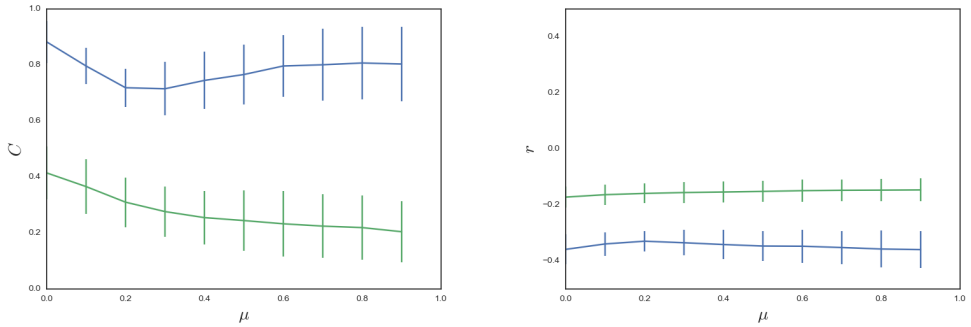
The construction and implicit assumptions about community structure made by the LFR approach are notably different to CiGRAM's. Most notably, CiGRAM is built under the conjecture that an underlying community is indistinguishable from a random graph and that the transitivity is a product of the block structure. LFR actually allows a configurable level of transitivity with a parameter that results in graphs with an approximate number of triangles. However, this is not mentioned in the article [128] and the general role of the community structure is determined to be the same; a random subgraph in this case is generated according to a fixed degree model.

A brief description of the construction procedure for LFR is as follows:

- Each node is assigned a degree from a power law distribution with exponent γ
- Nodes are assigned to communities, the sizes of which are drawn from a power law distribution with exponent κ.
- To allow overlap o_n nodes are assigned to o_m communities to allow overlap.
- The edges of the graph are assigned such that each node has a fraction $1 - \mu$ of its edges inside its assigned community and a fraction μ between communities.
- Rewiring is used to ensure that multiple edges do not occur

Figure 6.1 shows the behaviour of assortativity and the clustering coefficients with increasing μ in the LFR benchmarks with degree exponents $\gamma = 2.0$ and $\gamma = 2.8$. Similar results for CiGRAM are provided in Chapter 4. The clustering coefficient does not decrease predictably with μ and remains roughly constant across the models. Similarly, the models tested here appear to have disassortative structures, meaning that nodes have a propensity to connect to vertices with different degrees. This may be a product of the community structure or an inherent aspect of the model. However, the level of assortativity is not directly configurable.

One concerning aspect of the LFR benchmark is the fixed level of clustering across ranges of μ. In CiGRAM the clustering emerges only as a product of the



(a) Clustering *(b) Assortativity*

Figure 6.1: Clustering (a) and degree assortativity (b) coefficients of LFR benchmark models with increasing mixing μ. Models are generated with $\gamma = 2.0$ (blue) and $\gamma = 2.8$ (green) with community size distribution $\kappa = 1.4$. Each data point corresponds to the results taken from 32 realisations of these parameters.

community structure as random networks without communities lack any level of clustering. In this sense, fixing the number of triangles is a curious decision. Furthermore, it does not seem a fair test of module detection algorithms if the network generates dense subgraphs that are not considered communities.

6.3 Ensuring connectivity in CiGRAM

For graphs with relatively low density, there are no guarantees that CiGRAM will result in a single connected component. This presents a problem for the tests in this chapter, as internally, communities must be connected. To ensure that communities are internally connected, a rewiring procedure is defined that exploits the fact that, given an undirected graph $G = (V, E)$, the removal of any single edge contained within a cycle will not create a disconnected component. The proof for this trivially follows from the definition of a cycle. For an edge to be contained within a cycle there must be a path between all pairs of adjacent nodes $(i, j) \in E$ that do not include the adjacent edge. The removal of (i, j), therefore, cannot create a disconnected component as a path between i and j will still remain.

The rewiring procedure for CiGRAM is formally defined as follows. With a

set of edges E, we define the set of rewireable edges U as

$$U = \{(i,j) \in E | asp(i,j) < \infty\}, \tag{6.1}$$

where (i,j) is an edge in E and $asp(i,j)$ is the alternative shortest path between two nodes i and j when the direct edge (i,j) is excluded. The rewireable edges can be removed from E without creating disconnected components. If a graph, G, has one or more disconnected components they can be merged without changing the graph density by removing an edge $r \in U$ and adding edge r' between nodes in disconnected components. We select edge $r = (i,j)$ with probability

$$Pr(A_{ij} = 0) = 1 - (\beta_i \hat{\beta}(j|i) + \beta_j \hat{\beta}(i|j)), \tag{6.2}$$

the reader is referred to Chapter 4 Equation 4.9, which determines the weighted probability for each edge in the network. In essence, Equation 6.2 says that the least probable edges are removed first. U must be updated after each removal, and the number of possible further removals $U_p < |U|$.

We then select a first node from the largest connected component C_0, with probability

$$\beta_i = \frac{\alpha_i}{\sum_{u \in C_0} \alpha_u}, \tag{6.3}$$

and select a second node to form an edge, from the disconnected component C_1, with probability

$$\hat{\beta}(j|i) = \frac{\alpha_j e^{-a\delta(\theta_i,\theta_j)}}{\sum_{u \in C_1} \alpha_u e^{-a\delta(\theta_i,\theta_u)}}. \tag{6.4}$$

The reader is reminded of the Equation 4.14 in Chapter 4, where the distance between nodes i and j is defined as $\delta(\theta_i,\theta_u) = ||\theta_i| - |\theta_j||$ and a is the assortativity parameter. Such rewiring is only possible if there is a minimum of $n-1$ edges in the graph. Otherwise, additional edges have to be added to the graph and the target density is exceeded.

6.4 Assortativity and community structure

One aspect of networks not modelled by the LFR benchmark is that of positive and negative degree-degree correlations. In Chapter 4 Section 4.2.5 it was found that CiGRAM has distinct difficulty modelling assortativity when networks

become dense. Furthermore, the average clustering is high within networks with community structure. Under these locally dense configurations, assortativity is, again, difficult to model. Whilst it was not conclusively shown that this is an aspect of the model, rather than a property that is specific to networks, it supports the hypothesis that assortative graphs tend to be sparsely connected. This has strong implications for graphs with community structure. Either graphs have a large number of connections between communities (modelled by e_k in CiGRAM) that allow assortativity to remain high, or communities are, to some extent, sparsely connected internally.

Given that the basis for statistical approaches to community detection assumes that communities are densely internally connected, this has potential implications for their ability to correctly extract modular structure. This section describes a series of tests to evaluate the impact of degree-degree correlations upon the performance of module extraction algorithms. A notable limitation of CiGRAM is that the a parameter, which controls assortativity, also strongly impacts the degree distribution. For this reason we control the fit of the degree distribution to be as similar as possible across the range of assortative configurations.

The model parameters for community sizes $\tilde{\sigma}_f$ and $\tilde{\sigma}_s$ are fixed at 0.9, the density of the graphs is fixed at 0.02 and n is set at 500. The number of communities is set at $K = 10$. These parameters are fixed to allow CiGRAM to simultaneously model extremely heterogeneous degree distributions and a varying range of assortativity. A fitting procedure to linear increases in degree assortativity was designed. The objective is to fit the desired assortativity as well as the degree distribution; this is achieved through the particle swarm optimisation approach described in Chapter 5, Section 5.2. The above procedure was achieved for levels of e_k between 0.1 and 0.9. The target degree distribution is generated with parameters $\sigma_f = 1.2$, $\sigma_s = 0.8$ and $a = 1.5$ with the minimum degree set at 2. Networks fit the average cumulative distribution function (ACDF) of network degree taken from 1000 runs of the model with the above parameters.

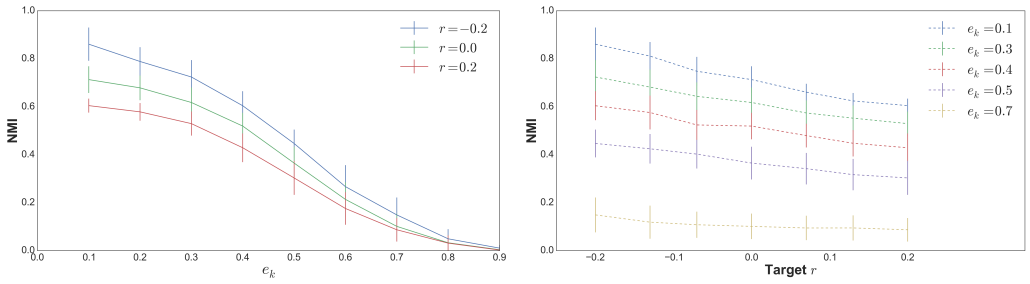
Appendix Figure C.1 demonstrates the level of fit achieved for the different target levels of r across the scales of e_k. Cumulative degree distribution and

complementary cumulative degree distribution plots are shown in Appendix Figures C.2 and C.3. The model error shown indicates 1 standard deviation for the KS distance found from 1000 samples of resulting degree distributions generated with the target set. All the resulting best fit model parameters appear to be within two standard deviations of the target degree distributions indicating that the degree distribution is controlled to be inside the model error for the target degree distribution. Similarly, the plots for the maximum degree are shown in Figure C.4. For comparison, the distribution of the maximum degree for the target model is shown in grey, indicating good fits across the ranges of e_k and r.

The fit for assortativity across the different ranges of the e_k parameter is shown in Figure C.5. These violin plots indicate the distribution of r across the range of best fit parameters. A major difficulty in achieving exact fits for assortativity appears to be the level of impact e_k has upon the resulting assortativity. For example, at $e_k = 0.1$ shown in Figure C.5 (a) the range of variance is extremely high, even though the distributions appear to be linear increases. As a consequence, the following results section ignores models generated outside the range of the target $r \pm 0.03$. As multiple samples are required this means re-sampling from the model parameters until the required number of sample graphs have been generated by the model.

6.4.1 Impact of assortativity results and discussion

The reader is reminded of the normalised mutual information (NMI) measure that first appeared in Chapter 3, Section 3.3.1. In this context, we are measuring the mutual information between a proposed partition or cover, found by a module section approach, against the ground-truth partition generated by CiGRAM. In order to show that a given algorithm's performance is significantly impacted by assortativity, the null hypothesis must be rejected. The null hypothesis can be stated formally as; the NMI scores for assortative, disassortative and non-assortative networks are drawn from the same distribution. In order for the null hypothesis to be accepted, the distribution of NMI scores must not be significantly lower in assortative configurations. This is tested by



(a) e_k response *(b) r response*

Figure 6.2: Normalised mutual information results on assortative graphs for the Infomap algorithm. (a) shows the response to increasing levels of e_k at fixed levels of r. (b) shows, for fixed levels of e_k, response to increases in r.

generating 100 sample networks and evaluating the performance of algorithms across different levels of assortativity. The algorithms tested in this section are all those listed in Chapter 3, Table 3.2.

Table 6.2 and 6.3 show the results of two sided Student's t-tests for the difference in the NMI scores at increasing levels of e_k for each module detection algorithm. If the p-value is greater than 0.01 the null hypothesis that the algorithm performs equally as well in the $r = 0$ model is not rejected, while it is rejected when the p-value is less than 0.01.

The Louvain, SA, Infomap and Hierarchical Infomap algorithms appear to be impacted by assortativity. The NMI scores are significantly lower for e_k values up to $e_k = 0.5$. Beyond $e_k > 0.5$ the NMI scores are very low indicating that the community structure is more difficult to detect, regardless of the level of assortativity. Interestingly, disassortativity appears to improve the performance of these algorithms. The NMI scores where $r = -0.2$ are significantly higher than those where $r = 0.0$ in all cases for these algorithms, with the exception of simulated annealing which shows no significant improvement in NMI scores at $e_k = 0.3$. These findings are presented in Figure 6.2, which shows the results for Infomap, increasing the level of e_k with fixed levels of r, and increasing the level of r with fixed levels of e_k. Additional results for the other algorithms are shown in Figures 6.4 and 6.5. The Louvain, Simulated Annealing and hierarchical Infomap algorithms show clear performance drops in the presence of assortative graphs not found in other algorithms.

Algorithm	r	$e_k = 0.1$ NMI	p	$e_k = 0.3$ NMI	p	$e_k = 0.5$ NMI	p	$e_k = 0.7$ NMI	p	$e_k = 0.9$ NMI	p
Louvain	-0.2	0.846	0.0	0.568	0.0	0.228	0.609	0.058	0.103	0.005	0.021
	0.0	0.7	1.0	0.521	1.0	0.232	1.0	0.049	1.0	0.002	1.0
	0.2	0.566	0.0	0.454	0.0	0.2	0.0	0.041	0.124	0.002	0.598
SA	-0.2	0.96	0.0	0.752	0.26	0.315	0.0	0.032	0.009	0.001	0.045
	0.0	0.923	1.0	0.729	1.0	0.258	1.0	0.019	1.0	0.0	1.0
	0.2	0.863	0.0	0.615	0.0	0.207	0.0	0.027	0.072	0.005	0.004
H. Infomap	-0.2	0.86	0.0	0.723	0.0	0.447	0.0	0.144	0.0	0.009	0.0
	0.0	0.713	1.0	0.615	1.0	0.363	1.0	0.101	1.0	0.004	1.0
	0.2	0.597	0.0	0.529	0.0	0.304	0.0	0.085	0.028	0.003	0.352
Lable Prop	-0.2	0.969	0.002	0.761	0.275	0.414	0.0	0.405	0.001	0.488	0.636
	0.0	0.98	1.0	0.742	1.0	0.335	1.0	0.316	1.0	0.482	1.0
	0.2	0.982	0.513	0.725	0.414	0.296	0.116	0.22	0.001	0.447	0.032
Infomap	-0.2	0.86	0.0	0.724	0.0	0.446	0.0	0.148	0.0	0.01	0.0
	0.0	0.713	1.0	0.617	1.0	0.365	1.0	0.101	1.0	0.003	1.0
	0.2	0.604	0.0	0.529	0.0	0.303	0.0	0.086	0.052	0.002	0.208
OSLOM	-0.2	0.933	0.314	0.733	0.969	0.312	0.0	0.1	0.246	0.248	0.024
	0.0	0.925	1.0	0.733	1.0	0.224	1.0	0.131	1.0	0.326	1.0
	0.2	0.919	0.434	0.689	0.023	0.221	0.88	0.136	0.854	0.119	0.0

Table 6.2: Significance of Normalised Mutual Information (NMI) scores as a measure for the ability to recall true modules in graphs generated with CiGRAM at varying levels of assortativity. Level of assortativity is indicated by r (see Equation 2.26). p-values are from the Student's t-test between the distribution of NMI scores and the distribution of values where $r \approx 0.0$.

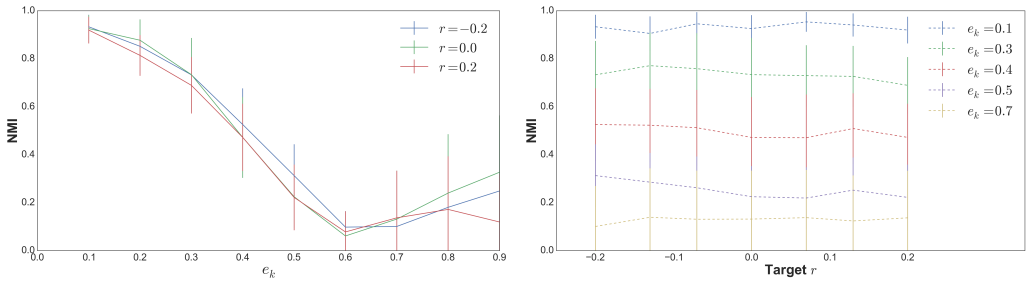
Algorithm	r	$e_k = 0.1$ NMI	p	$e_k = 0.3$ NMI	p	$e_k = 0.5$ NMI	p	$e_k = 0.7$ NMI	p	$e_k = 0.9$ NMI	p
COPRA $v = 1$	-0.2	0.682	0.0	0.516	0.024	0.242	0.0	0.192	0.0	0.216	0.195
	0.0	0.845	1.0	0.569	1.0	0.115	1.0	0.052	1.0	0.175	1.0
	0.2	0.923	0.0	0.579	0.664	0.096	0.198	0.034	0.033	0.146	0.332
COPRA $v = 2$	-0.2	0.686	0.0	0.549	0.062	0.422	0.663	0.349	0.477	0.238	0.279
	0.0	0.831	1.0	0.595	1.0	0.413	1.0	0.326	1.0	0.275	1.0
	0.2	0.887	0.006	0.57	0.289	0.348	0.004	0.273	0.103	0.328	0.112
COPRA $v = 3$	-0.2	0.747	0.0	0.493	0.956	0.478	0.651	0.378	0.065	0.237	0.0
	0.0	0.879	1.0	0.494	1.0	0.473	1.0	0.427	1.0	0.379	1.0
	0.2	0.922	0.005	0.521	0.246	0.425	0.005	0.402	0.304	0.376	0.935
COPRA $v = 4$	-0.2	0.828	0.007	0.465	0.1	0.458	0.215	0.417	0.122	0.284	0.003
	0.0	0.874	1.0	0.437	1.0	0.435	1.0	0.377	1.0	0.376	1.0
	0.2	0.902	0.047	0.455	0.336	0.443	0.695	0.366	0.684	0.361	0.615
COPRA $v = 5$	-0.2	0.807	0.052	0.478	0.179	0.488	0.084	0.483	0.316	0.364	0.001
	0.0	0.845	1.0	0.461	1.0	0.468	1.0	0.469	1.0	0.452	1.0
	0.2	0.873	0.074	0.43	0.057	0.427	0.026	0.411	0.005	0.424	0.209
COPRA $v = 6$	-0.2	0.795	0.301	0.447	0.723	0.5	0.016	0.49	0.266	0.434	0.066
	0.0	0.817	1.0	0.452	1.0	0.483	1.0	0.498	1.0	0.47	1.0
	0.2	0.848	0.088	0.432	0.192	0.469	0.266	0.483	0.078	0.469	0.99
COPRA $v = 7$	-0.2	0.749	0.017	0.457	0.875	0.5	0.055	0.5	1.0	0.443	0.133
	0.0	0.805	1.0	0.459	1.0	0.49	1.0	0.5	nan	0.471	1.0
	0.2	0.817	0.51	0.438	0.171	0.479	0.259	0.487	0.083	0.479	0.575

Table 6.3: Significance of Normalised Mutual Information (NMI) scores as a measure for the ability to recall true modules in graphs generated with CiGRAM at varying levels of assortativity. Level of assortativity is indicated by r (see Equation 2.26). p-values are from the Student's t-test between the distribution of NMI scores and the distribution of values where $r \approx 0.0$.

The decrease in performance in the presence of high assortativity with these algorithms may be explained by the nature of their composition. The modularity based Louvain and simulated annealing algorithms measure the significance of a graph using the configuration model as a null comparison (see Equation 2.11). This null model does not include a notion of assortativity, and this may have an impact on the results. Similarly, the Infomap algorithms use the degree of nodes to compute the transition probabilities for random walkers but ignores any correlation between them. In the case of a network with positive assortativity, the probability of transitioning to a node of similar degree is significantly higher. The definition in Chapter 2, Section 2.4.3 does not include this behaviour, indicated that it is not considered as part of the partition quality measure.

Under Student's t-test, the OSLOM algorithm shows no statistically significant impact from the inclusion of assortativity or disassortativity. The distribution of NMI scores does not reject the null hypothesis that the NMI scores are significantly lower for graphs with higher levels of assortativity. Though similar in conception to the modularity based algorithms through the use of the statistical significance clusters, there are several major differences, explained in more detail in Chapter 2, Section 2.4.4. Most notably, the implementation of OSLOM included here uses consensus clustering based on multiple runs of the algorithm, giving the results of a median cover rather than a single run. Furthermore, the notion of a community includes statistical significance starting from a seed node and expanding until clear modular structure is observed. Thus, the assumed null model is based on observed topological properties beyond the degree distribution. In contrast, modularity maximisation finds unlikely communities, but makes no judgement about their statistical significance. The results for the OSLOM algorithm are shown in Figure 6.3. When contrasted with the results for Infomap in Figure 6.2, it is particularly clear that assortativity is not causing a significant impact on the performance of OSLOM.

The label propagation method, first described in Chapter 2, Section 2.4.5, appears to have statistically significant fluctuations in NMI scores across the ranges of e_k for disassortativity, by measure of the p-value under the Student's



(a) e_k response *(b) r response*

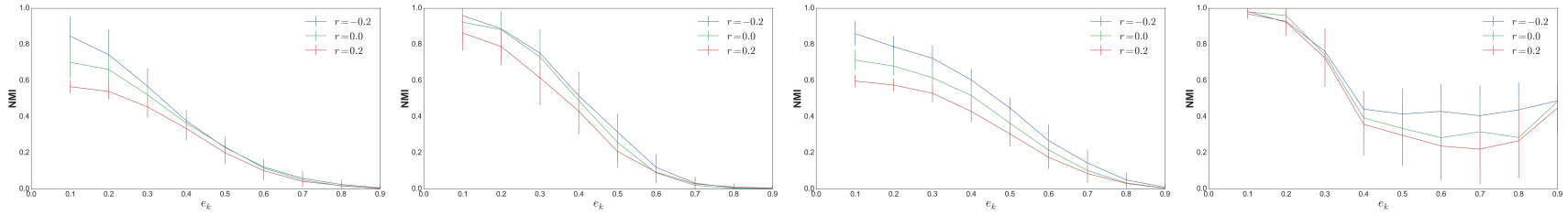
Figure 6.3: Normalised mutual information results on assortative graphs for the OSLOM algorithm. (a) shows the response to increasing levels of e_k at fixed levels of r. (b) shows, for fixed levels of e_k, response to increases in r.

t-test. However, these results lack consistency across the range of e_k, making it difficult to argue that this supports the hypothesis that label propagation performs better in disassortative graphs. Furthermore, Label propagation is also not significantly impacted by increases in the level of assortativity as the distributions of NMI scores for assortative and non-assortative configurations are not significantly different under the Student's t-test (see Table 6.2).

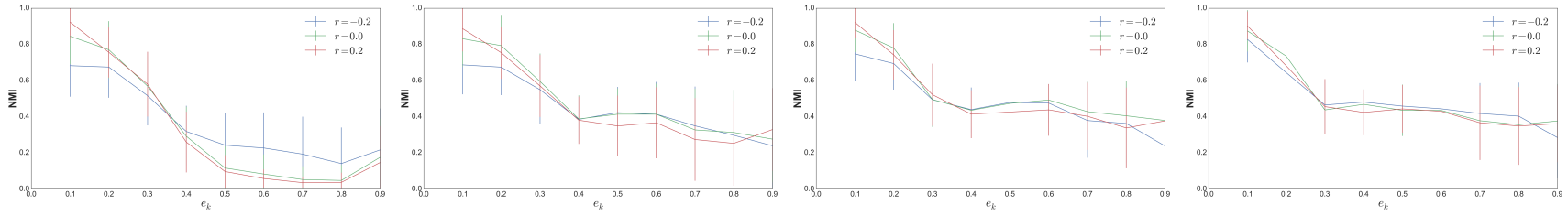
Interestingly, the results of COPRA algorithms, shown in Table 6.3, never fall below an NMI score of around 0.1, indicating good average performance in response to e_k. Moreover, the results in this table indicate that assortativity does not appear to have a significant impact upon the results of the algorithm. Notably, at $e_k = 0.1$, the performance in assortative configurations is actually better than $r = 0$ configurations or $r = -0.2$ configurations. The reasons for this are unknown, and where $v > 3$ this does not appear to be the case.

6.5 Benchmarks for algorithm selection

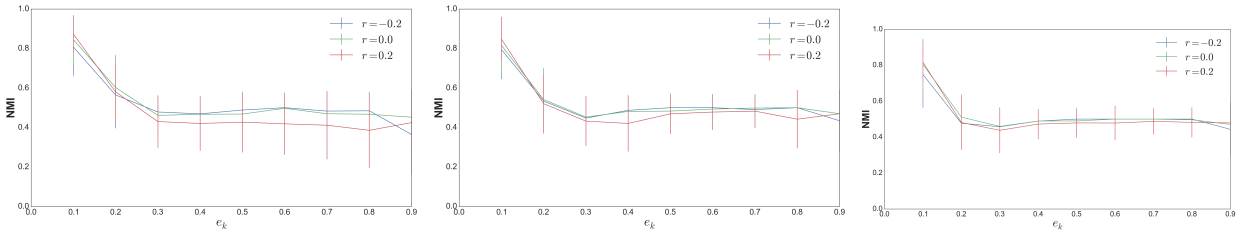
The results of previous sections indicate that selecting the best community detection algorithm depends on a large number of competing properties. Given that the topology of empirical datasets differs massively, it is extremely unlikely that a single algorithm performs well on any given graph. Previous work into benchmark graphs attempts to use universal properties such as scale-free degree distributions to rank the selection of algorithms in a universal manner [159,196]. However, Chapter 3 showed a complete lack of consensus between different



(a) Louvain *(b) SA* *(c) H Infomap* *(d) Label Propagation*

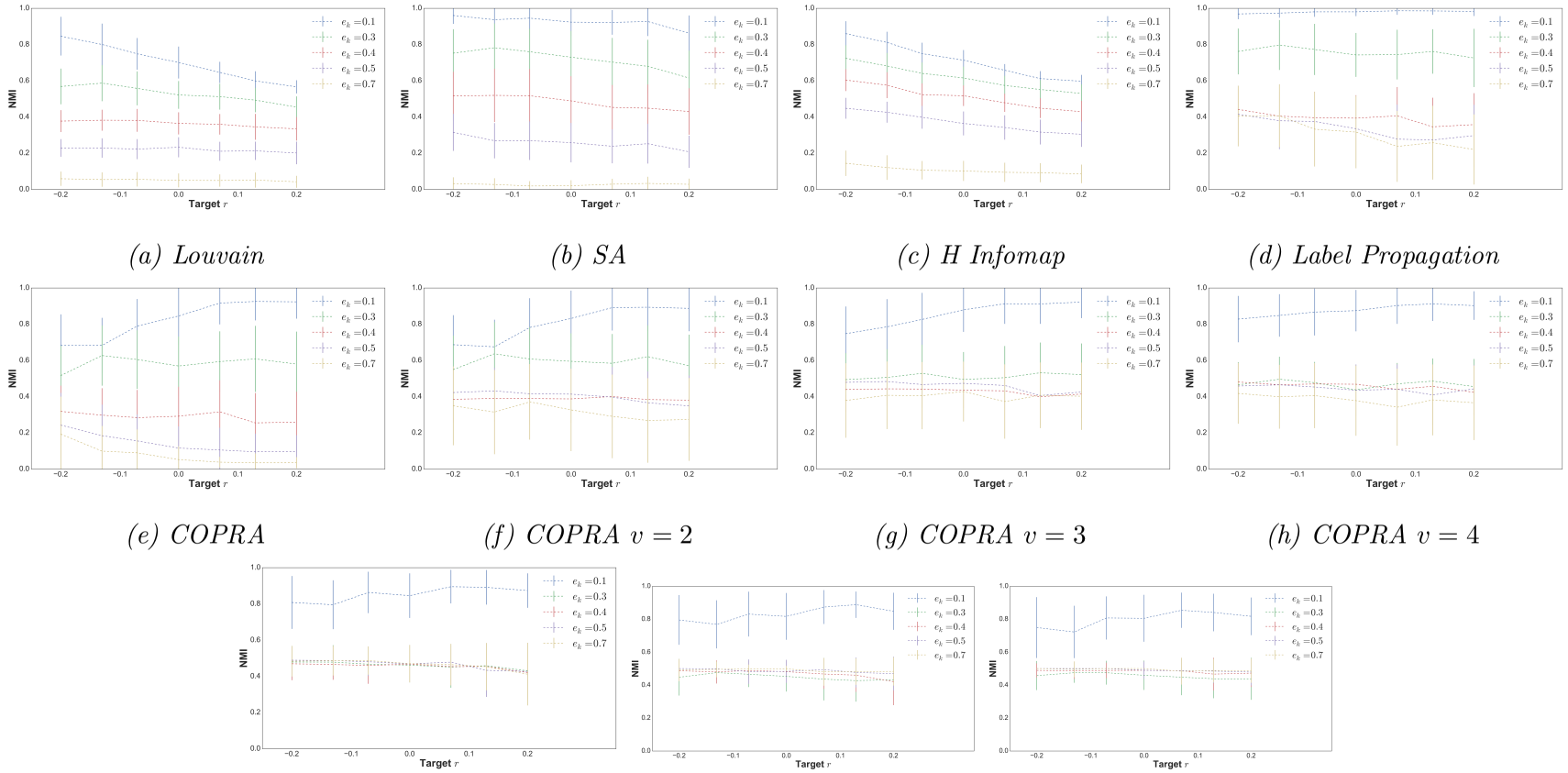


(e) COPRA *(f) COPRA $v=2$* *(g) COPRA $v=3$* *(h) COPRA $v=4$*



(i) COPRA $v=5$ *(j) COPRA $v=6$* *(k) COPRA $v=7$*

Figure 6.4: Normalised mutual information results on assortative graphs for different algorithms. Shows the response to increasing levels of e_k at fixed levels of assortativity, r.



(a) Louvain *(b) SA* *(c) H Infomap* *(d) Label Propagation*

(e) COPRA *(f) COPRA $v = 2$* *(g) COPRA $v = 3$* *(h) COPRA $v = 4$*

(i) COPRA $v = 5$ *(j) COPRA $v = 6$* *(k) COPRA $v = 7$*

Figure 6.5: Normalised mutual information results on assortative graphs for different algorithms. Results compare, for fixed levels of e_k, response to NMI scores against increases in assortativity, r.

methods. Since detected modules may be used to generate biological hypotheses, a well-grounded methodology when selecting module extraction methods is clearly needed.

In this section, we define a methodology for selecting an algorithm used on real world networks that is based on the use of benchmark graphs, a core objective of the thesis. For our purposes, CiGRAM is the benchmark graph used. However, this should not preclude the use of other benchmarks such as the LFR or the BTER models [134]. The objective is to test algorithms in context specific manner. In other words, given the best available representation of a topological structure, with a known, configurable community structure, which algorithm performs best on this synthetic dataset? It is both unrealistic and unreasonable to assume an algorithm will perform well in all circumstances given the complexity and variety of empirical data.

The outline of the methodology for algorithm selection can be described as follows:

1. **Selection of appropriate synthetic benchmark models and parameters.** The synthetic benchmark graph selected should provide a representation of the real world network that is as realistic as possible. This means that topological features such as clustering, assortativity and degree distributions should closely resemble the target graph. Chapter 5 explored two approaches to fitting networks: manually matching summary statistics of graphs and using the distance between graph spectra. Neither choice was adequate for representing all topological properties of the real world graph and all models will have some degree of inaccuracy. However, the use of a test on a synthetic dataset is always better than no evaluation, and multiple fits is better than a single fit.

2. **Test algorithms under a wide range of conditions.** Real data is prone to noise. Therefore, the algorithms should be tolerant to a higher level of overlap and mixing between communities than is expected in the real graph. In CiGRAM this is achieved by varying the e_k and p_o parameters and measuring the response in NMI scores between the algorithms.

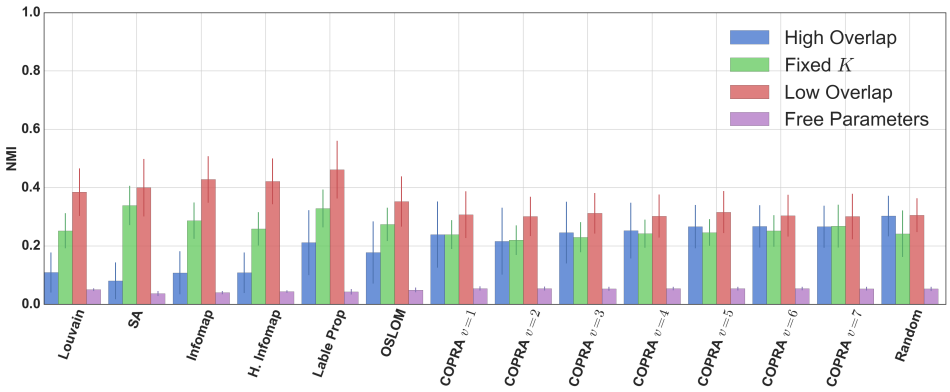


Figure 6.6: Performance of community detection algorithms in terms of Normalised Mutual Information of 32 realisations of the best fit E coli metabolic network model graphs from Chapter 5. Error bars indicate standard deviation.

3. **Selection of best algorithm or algorithms.** The algorithm or algorithms with the consistently highest performance across the range of test models should be selected. Where ambiguity is found, the selection of multiple algorithms should be considered by comparing the consensus of multiple clusterings. This may not be achieved with a single algorithm and additional results such as meta-data can be used for further validation.

The remainder of this section uses the example networks from Chapter 5, highlighting how the above methodology can be used.

6.5.1 Performance on best fit graphs

This section discusses the performance of module detection algorithms on the best fit models of the real world datasets from Chapter 5. In this section, each model is generated 32 times, with the ground-truth performance evaluated on each model. A larger number of replicates would, naturally, give more confidence in the results. However, a sample size of 32 was chosen in view of the length of time taken to compute the clustering on some of the larger network models, whilst still being large enough to accurately capture the standard deviation of the algorithm performance across the model variance. As with Section 6.4.1, the algorithms tested in this section are listed in Chapter 3, Table 3.2.

Figures 6.6, 6.7 and 6.8 show the performance of algorithms on the best fit graphs of biological and non-biological models, respectively. These plots show the mean NMI scores between the clusters extracted by each algorithm and the ground-truth clusters generated by CiGRAM (see Chapter 3, Section 3.3.1 for a definition of NMI). Each of the bars relates to one of the best fit models under the different conditions described in detail in Section 5.5.3. The error bars indicate the standard deviation of the performance. In addition to the model results, the mean NMI score for 100 random partitions is compared to the performance of the algorithms providing a baseline level of performance to compare results against. The random partitions are generated by generating a random cut set. It should be noted that this has a bias towards edges not contained within cycles. Consequently, the random partitions score higher NMI scores where the fraction of edges between communities is lower (e.g. edges between communities are less likely to be contained within cycles). However, the partitions generated are not generated through any form of optimisation.

The plots show no clear, consistent best algorithm across all the networks, and there is considerable variation in mutual information scores between the networks. In order to more formally quantify these results, the aggregate ranking for the scores shown in the biological networks is given in Table 6.4 for the biological network and Table 6.5 for other networks. The score is taken from the mean of the normalised mutual information scores across the range of tests a ranking is defined as

$$\text{score} = \frac{1}{|T|} \sum_{P' \in T} NMI(P, P'), \tag{6.5}$$

where T is the set of all solutions to all tests, P' is a solution in T and P is the ground-truth solution. Equation 6.5 is the mean normalised information across all clusterings for each test, including all 32 replicates for each best fit model. The reason an aggregate score is used is because the a given algorithm may perform well under certain conditions, such as low overlap, but poorly in another condition. As the models are certainly inaccurate a given algorithm should perform well across the range of test graphs.

The ranking in Tables 6.4 and 6.5 also includes p-values. The data are from aggregates over different models and cannot be assumed to be normally

distributed, this makes the one sided, non-parametric Mann-Whitney U test suitable. The test is conducted between each algorithm and the algorithm ranked lower than it. In order for the performance to be considered significantly better than the algorithm ranked below, the null hypothesis that the median of the distributions are the same must be rejected. Here we state that $p < 0.01$ rejects the null hypothesis that the two distributions are identical.

The reader should note, however, that the adoption of a given p-value does not mean that the algorithm should be excluded from further analysis. For, if the aggregate NMI performance of a given algorithm is 0.9 with the next best algorithm having a performance of 0.89, it would be wise to include both algorithms in further analysis, regardless of the p-value. Formally the null-hypothesis states that, given two distributions X and Y there is a 50% chance of drawing a value taken from X in the distribution Y. Label propagation appears to rank consistently well across all tests on all networks. However, it is important to point out that, with the exception of SeedNet, none of the networks perform particularly highly on any of the benchmark networks. This result is in contrast with benchmarks conducted on the LFR networks [159, 196] which show algorithm performance with average NMI scores consistently close to 1 for many of the algorithms tested here.

The results of the combined NMI scores are exceptionally very poor in the case of the *E coli* metabolic network models. In all other cases most of the algorithms perform consistently better than a random graph. However, for the *E coli* network, shown in Figure 6.6, only label propagation performs better than random. Even in this case, the performance is only 15% better than the random result with a score of 0.226. Furthermore, this is not simply skew from a single model class; the result is consistent across each of the cases. The reason for the poor performance in the case of the *E coli* models is unclear.

In other cases of models, the performance of algorithms is equally as bad but the aggregate results in Tables 6.4 and 6.5 do not reflect this. For example, the fixed high and low overlap results for the Yeast PPI models show that none of the algorithms are better than random chance, a result repeated in the Open Flights network in the context of low overlap. Similarly, the results are extremely poor on the PGP models, again, with many of the algorithms

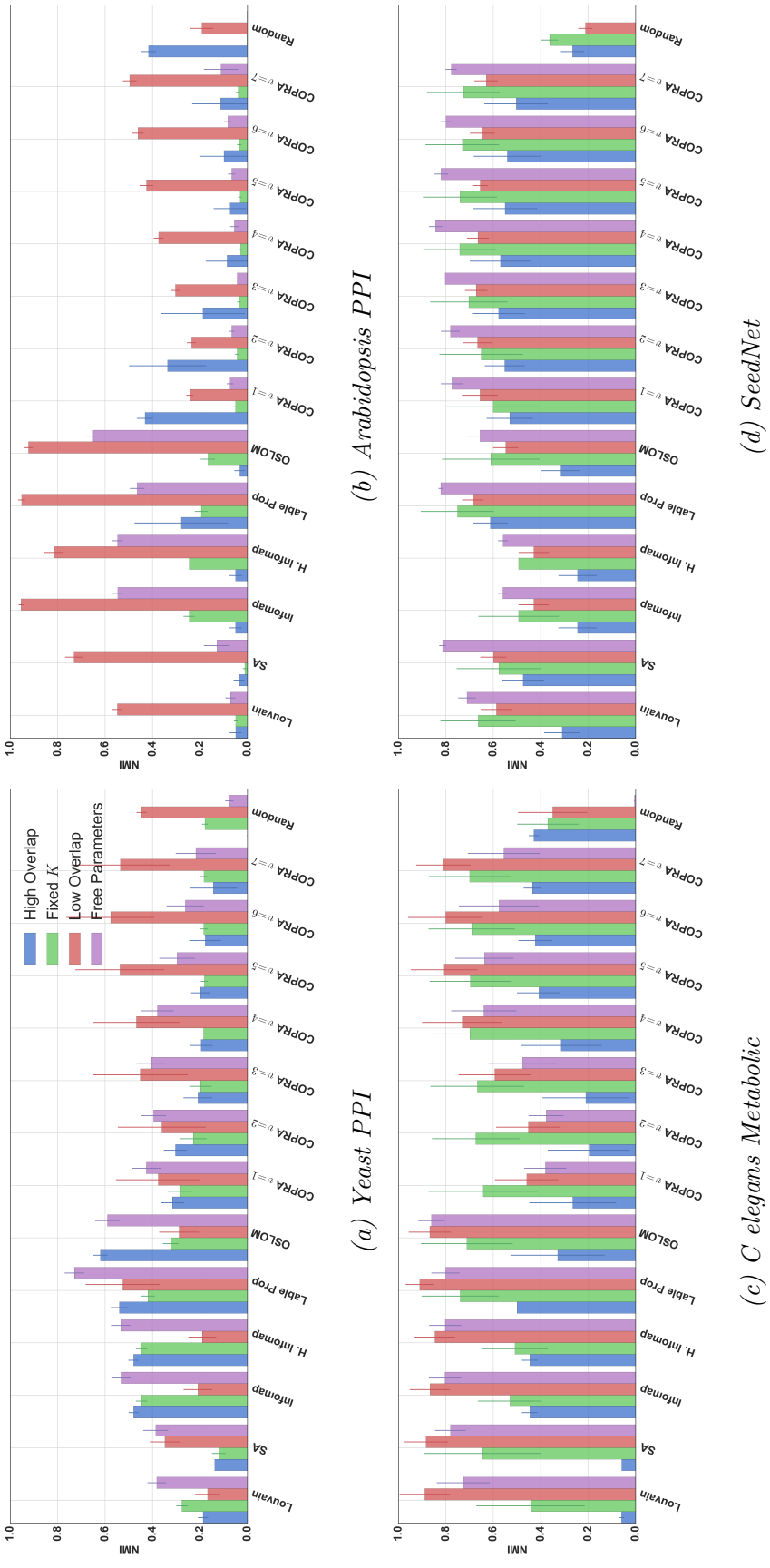
appearing to perform better than random across the different test cases.

Further performance evaluation

A naive assumption would be that the poor performance correlates directly with mixing, e_k. Whilst e_k and p_o do have an impact on the performance of algorithms, results comparing the NMI scores do not indicate that this is the case. Figure 6.9 plots the mean best algorithm NMI scores on each network against the e_k levels assigned by the optimisation process in Chapter 5. There appears to be no correlation between the performance of the algorithms and e_k. This indicates that other factors must influence the performance of algorithms. For example, the US power grid is extremely sparse, with a high level of clustering relative to its edge density. This means that there is a large number of modules that vary in size, a property known to cause difficulty for modularity maximisation approaches [86].

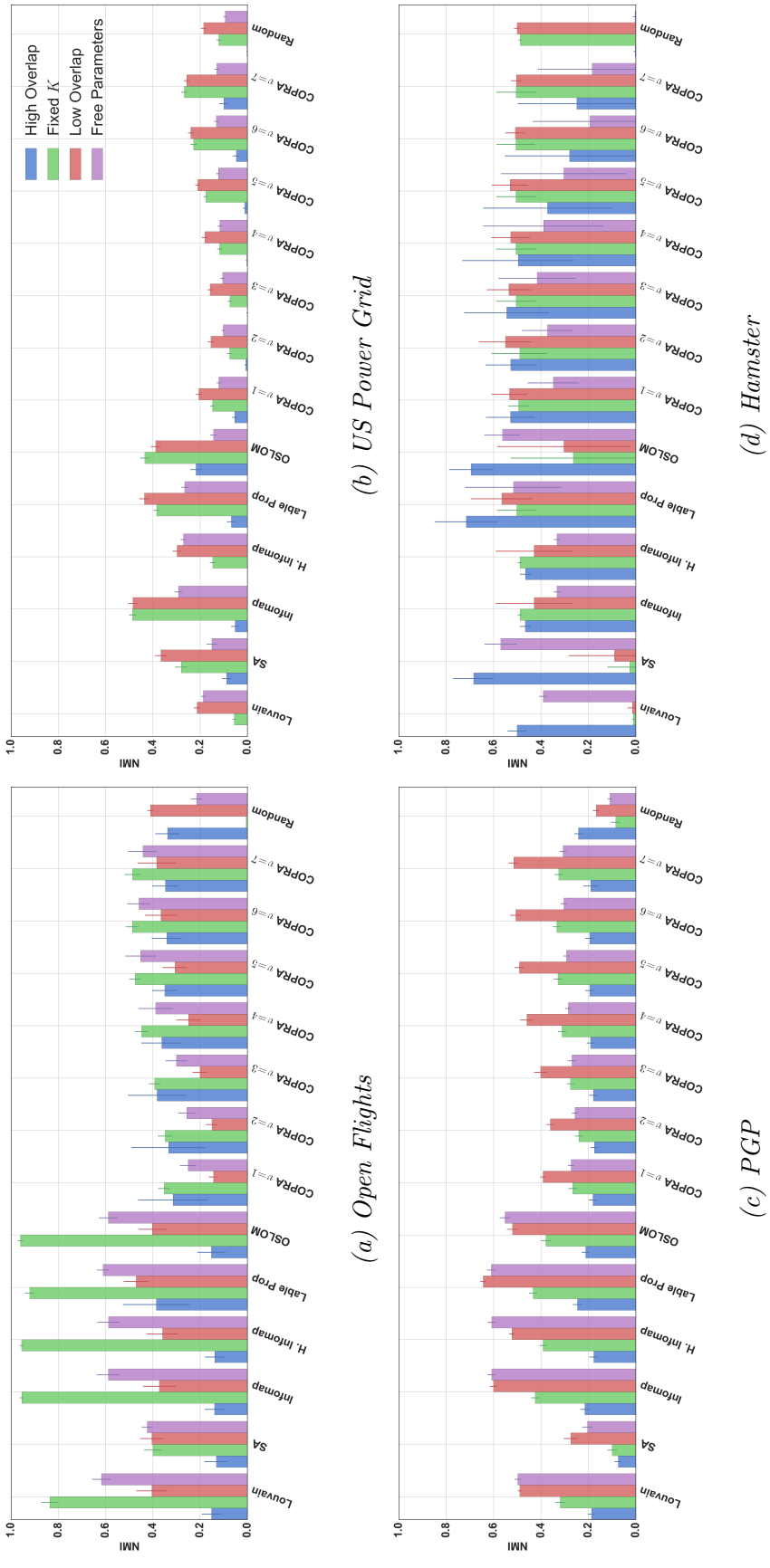
One possible explanation for the poor performance of algorithms on these benchmarks is that the community structure detected by the algorithms is actually higher quality solution than the K blocks generated by CiGRAM. In this case, one would expect to see a high level of consistency between the algorithms. In other words, the NMI scores of solutions proposed by different algorithms should be high. For example, it may be that a given partition has high modularity but virtually no similarity to the true set of clusters. However, in the case of Good et al [81], the modularity search space was shown to be extremely glassy, indicating that real networks have many locally optimal solutions that are extremely dissimilar in terms of mutual information. In order to test this hypothesis we use a similar approach as Chapter 3, Section 3.3.1, shown visually in Figure 3.2.

We show example heat plot results of the similarity across the different algorithms in Figure 6.10 for the free parameter benchmark graphs of the biological datasets. Appendix Figures C.6 to C.8 show similar results for the other model conditions. These plots show the level of NMI between the solutions found by a subset of the algorithms, taking the mean score from the 32 replicates for each benchmark model. Only COPRA with $v = 5$ is included and the hierarchical form of Infomap is excluded. The reason for this is that



(a) Yeast PPI *(b) Arabidopsis PPI* *(c) C elegans Metabolic* *(d) SeedNet*

Figure 6.7: Performance of community detection algorithms in terms of Normalised Mutual Information of 32 realisations of the best fit of biological graphs from Chapter 5. Error bars indicate standard deviation.



(a) Open Flights *(b) US Power Grid* *(c) PGP* *(d) Hamster*

Figure 6.8: Performance of community detection algorithms in terms of Normalised Mutual Information of 32 realisations of the best fit of non-biological graphs from Chapter 5. Error bars indicate standard deviation.

SeedNet	score	p-value	Yeast PPI	score	p-value	C Elegans	score	p-value	E coli	score	p-value	Arabidopsis PPI	score	p-value
Label Prop	0.718	0.318	Lable Prop	0.554	0.0	Lable Prop	0.738	0.305	Lable Prop	0.261	0.075	Lable Prop	0.473	0.469
COPRA $v = 4$	0.705	0.208	OSLOM	0.456	0.006	OSLOM	0.692	0.008	Random	0.226	0.355	Infomap	0.449	0.138
COPRA $v = 5$	0.692	0.368	Infomap	0.417	0.453	Infomap	0.661	0.314	COPRA $v = 7$	0.222	0.481	OSLOM	0.444	0.31
COPRA $v = 3$	0.689	0.275	H. Infomap	0.413	0.0	H. Infomap	0.651	0.206	COPRA $v = 5$	0.22	0.494	H. Infomap	0.415	0.0
COPRA $v = 6$	0.679	0.139	COPRA $v = 1$	0.35	0.012	COPRA $v = 5$	0.637	0.361	COPRA $v = 6$	0.219	0.134	SA	0.226	0.0
COPRA $v = 2$	0.663	0.421	COPRA $v = 2$	0.322	0.072	COPRA $v = 7$	0.625	0.376	Infomap	0.216	0.372	COPRA $v = 1$	0.2	0.19
COPRA $v = 7$	0.659	0.206	COPRA $v = 3$	0.316	0.238	COPRA $v = 6$	0.623	0.334	SA	0.214	0.235	COPRA $v = 7$	0.19	0.166
COPRA $v = 1$	0.64	0.117	COPRA $v = 4$	0.307	0.179	COPRA $v = 4$	0.596	0.037	OSLOM	0.213	0.463	Louvain	0.18	0.153
SA	0.616	0.067	COPRA $v = 5$	0.303	0.181	SA	0.592	0.051	COPRA $v = 4$	0.213	0.45	COPRA $v = 2$	0.171	0.278
Louvain	0.568	0.016	COPRA $v = 6$	0.301	0.067	Louvain	0.529	0.026	COPRA $v = 3$	0.21	0.49	COPRA $v = 6$	0.169	0.0
OSLOM	0.532	0.0	COPRA $v = 7$	0.27	0.083	COPRA $v = 3$	0.487	0.005	COPRA $v = 1$	0.21	0.118	Random	0.153	0.003
Infomap	0.432	0.477	Louvain	0.253	0.167	COPRA $v = 1$	0.437	0.316	H. Infomap	0.208	0.379	COPRA $v = 5$	0.149	0.185
H. Infomap	0.431	0.0	SA	0.249	0.0	COPRA $v = 2$	0.425	0.0	Louvain	0.199	0.305	COPRA $v = 3$	0.143	0.483
Random	0.21	-	Random	0.176	-	Random	0.289	-	COPRA $v = 2$	0.198	-	COPRA $v = 4$	0.136	-

Table 6.4: Overall performance of algorithms ranked by mean NMI scores across all test models and samples for the biological networks from Chapter 5. This table aggregates the results shown in Figures 6.6 and 6.7. The p-values in this table are taken from the one sided Mann-Whitney U test and compare the distribution of NMI scores for the algorithm against the algorithm below. Here, the null hypothesis is that there is a 50% chance that an NMI score from one algorithm would be found by the other algorithm. Where $p < 0.01$ we reject the null hypothesis that any difference in scores is due to natural variation in an identical distribution.

Open Flights	score	p-value	US Power Grid	score	p-value	Hamster	score	p-value	PGP	score	p-value
Label Prop	0.597	0.037	Infomap	0.329	0.0	Lable Prop	0.575	0.0	Lable Prop	0.484	0.003
OSLOM	0.526	0.249	OSLOM	0.296	0.398	COPRA $v = 3$	0.501	0.118	Infomap	0.462	0.002
Infomap	0.514	0.464	Lable Prop	0.288	0.0	COPRA $v = 2$	0.485	0.038	H. Infomap	0.425	0.229
H. Infomap	0.51	0.468	SA	0.221	0.004	COPRA $v = 4$	0.48	0.001	OSLOM	0.416	0.0
Louvain	0.503	0.001	COPRA $v = 7$	0.188	0.035	COPRA $v = 1$	0.476	0.009	Louvain	0.374	0.036
COPRA $v = 7$	0.415	0.488	H. Infomap	0.179	0.001	OSLOM	0.457	0.0	COPRA $v = 6$	0.335	0.465
COPRA $v = 6$	0.413	0.054	COPRA $v = 6$	0.161	0.0	Infomap	0.429	0.453	COPRA $v = 7$	0.335	0.151
COPRA $v = 5$	0.396	0.003	COPRA $v = 1$	0.132	0.33	H. Infomap	0.429	0.0	COPRA $v = 5$	0.327	0.052
COPRA $v = 4$	0.362	0.295	COPRA $v = 5$	0.13	0.149	COPRA $v = 5$	0.428	0.023	COPRA $v = 4$	0.312	0.001
SA	0.34	0.006	Louvain	0.114	0.054	COPRA $v = 6$	0.372	0.33	COPRA $v = 3$	0.282	0.304
COPRA $v = 3$	0.319	0.0	COPRA $v = 4$	0.105	0.1	COPRA $v = 7$	0.361	0.095	COPRA $v = 1$	0.278	0.001
COPRA $v = 2$	0.272	0.361	Random	0.1	0.015	SA	0.343	0.0	COPRA $v = 2$	0.258	0.0
COPRA $v = 1$	0.265	0.464	COPRA $v = 2$	0.085	0.371	Random	0.249	0.119	SA	0.163	0.311
Random	0.242	-	COPRA $v = 3$	0.085	-	Louvain	0.229	-	Random	0.151	-

Table 6.5: Overall performance of algorithms ranked by mean NMI scores across all test models and samples for the non-biological networks from Chapter 5. This table aggregates the results shown in Figure 6.8. The p-values in this table are taken from the one sided Mann-Whitney U test and compare the distribution of NMI scores for the algorithm against the algorithm below. Here, the null hypothesis is that there is a 50% chance that an NMI score from one algorithm would be found by the other algorithm. Where $p < 0.01$ we reject the null hypothesis that any difference in scores is due to natural variation in an identical distribution.

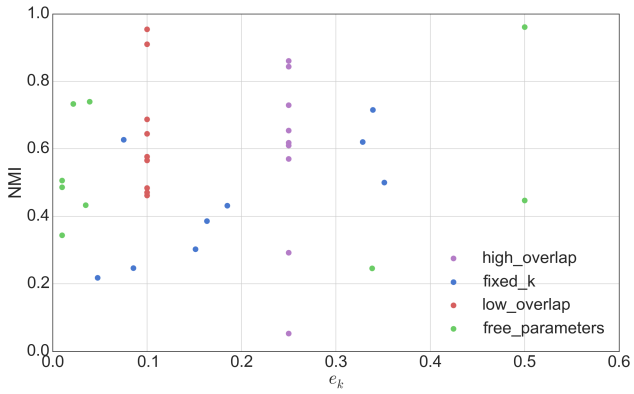


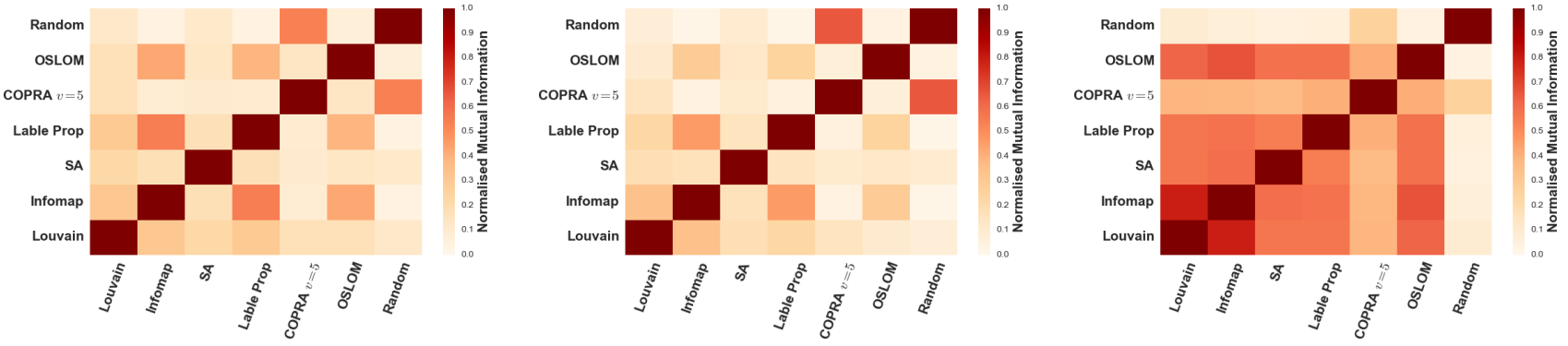
Figure 6.9: Best Normalised mutual information results from Figures 6.6, 6.7 and 6.8 against levels of e_k assigned by the optimiser in Chapter 5.

previous results in Chapter 3 give us reason to believe that the algorithms will perform consistently across the datasets.

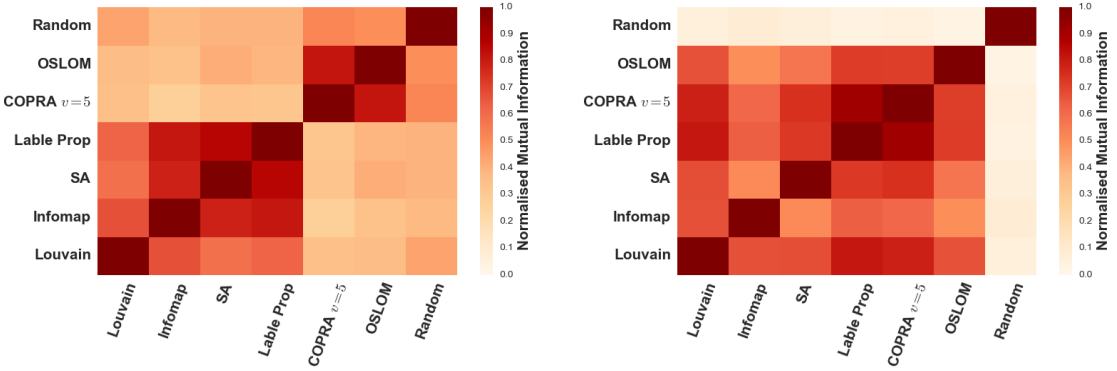
These plots show that, whilst the results are more similar to one another than would be expected for random clusterings, the clusterings rarely achieve high levels of agreement. The *E coli* metabolic network shows high similarity between label propagation, simulated annealing, Louvain and Infomap, but disagreement with COPRA and OSLOM. The solutions generated for the *E coli* metabolic network have, broadly, more similarity with one another than compared with random clusterings. However, there is no clear form of consensus, matching results from analysis in [81] (described in Chapter 2 Section 2.4.2) that highlighted a highly glassy optimisation landscape. In the case of the SeedNet models, the algorithms perform more consistently. This can be explained by the fact that the performance across these algorithms is quite good, with results above 0.8 for many of the algorithms.

6.5.2 Performance summary

This section has measured the performance of popular module detection methods when applied to ground-truth modular structures generated by CiGRAM that aim to closely match the topology observed in real world data. Surprisingly, the community extraction methods perform poorly in a number of situations. This may be due to the benchmark models being too strict and generating topology that is too hard to discover. However, in the case of the *E coli* models,



(a) Yeast PPI *(b) Arabidopsis PPI* *(c) C elegans Metabolic*



(d) E coli Metabolic *(e) SeedNet*

Figure 6.10: Normalised mutual information consensus matrix for agreement between algorithms on five best fit biological networks using the free parameters modelling approach described in Chapter 5 Section 5.5.3.

many of these approaches perform no better than random indicating further analysis of these algorithms is required. Furthermore, there appears to be no correlation between the e_k parameter and the performance of the algorithms when considering the range of the values. As with the previous section regarding assortativity, the poor performance appears to be a consequence of the underlying network topology.

This process presents an approach to selecting an algorithm on the real world datasets. The lack of assumptions made by the Label Propagation algorithm and the accuracy of the OSLOM algorithms appear to make them, quantitatively, the best choices. It should be noted, however, that this method should not replace the use of meta-data such as the gene ontology and experimental data covered in Chapter 3.

6.6 Chapter summary

This chapter has focused on evaluating the performance of module detection algorithms in the context of real world structure. Firstly, in Section 6.4.1, the performance of algorithms in relation to degree assortativity was evaluated. Following this, a methodology for the selection of community detection algorithms was discussed.

In Section 6.4.1, it was discovered that some algorithms appear to perform worse in networks with high levels of assortativity, controlling for the degree distribution and the coherence of modules in the form of edges between communities. These results have implications for both algorithm designers and those that wish to use module detection algorithms in a practical context. Modularity maximisation, for example, uses a form of null model that does not consider assortativity to be relevant when considering the statistical significance of modular structures. Similarly, algorithms such as OSLOM and label propagation appear to be unimpaired by increased levels of assortativity, indicating that these algorithms may be a better choice for empirical networks with this topology.

The results in Section 6.5.1 show that algorithm selection is not a trivial process and depends heavily upon the topology of the observed networks.

Whilst some generalisations about community structure can be made from first principles, in a practical context there is no single universal solution for module detection. Whilst the quality of performance of many of these algorithms appears to be low, in this context, it is worth remembering that this does not mean that work that uses these clusters is invalid. In Chapter 3 it was found that many of these algorithms find statistically significant results in real world datasets. Simply because the normalised mutual information scores here are low does not mean that core clusters, such as cliques and other statistically rare dense groups are not related.

Some criticism of the methodology for algorithm selection should be noted, the core of which should focus on model selection. Whatever model one chooses to represent a real world topology, there will always be some level of disagreement between empirical data and those selected. Fundamentally, the geometric approach to modelling probability spaces taken in this thesis makes the underlying assumption that random graphs create no modular structure. Future models and work may improve upon or reject this assumption but the selection of algorithms, where no reliable ground-truth data are available, must still rely on accurate representations of real world graph topology.

Chapter 7

Conclusions and future work

7.1 Thesis summary

The core aim of this thesis has been to provide a methodology to evaluate the performance of module extraction methods in the context of realistic topology. To frame this work in the context of the field, the thesis explored module detection approaches in complex biological networks through modelling topological structure observed in empirical datasets. his led to a stronger definition of a modular graphs, allowing a ground-truth model for evaluating module extraction methods. Fitting this model to real world datasets allowed the development of formal methodology for algorithm selection in domains with inaccurate meta-data for underlying modules.

The isolation of modules has been highlighted as an important method for the generation of biological hypotheses [3]. In Chapter 3, however, the different approaches to module extraction show very little similarity with one another when compared by measure of mutual information and gene ontology detected. In the *Arabidopsis thaliana* networks, it was shown that the modules identified relate to clusters of genes that are evolutionarily conserved, supporting a previous hypothesis [142] that certain stages of embryo-genesis are crucial to plant development. The potential for module detection in a biological context is massive. However, methods for statistically validating topological structure must be developed to improve confidence in results.

Chapter 4 presented CiGRAM, a novel model for the generation of undirected graphs. It was shown that the modification of latent variables and score

functions through use of wrapped normal distributions can generate a very wide range of degree heterogeneity. Furthermore, the use of hidden geometric variables allows the generation of graphs with positive and negative degree-degree correlations (assortativity and disassortativity). The generation of modular structure follows from the definition that a module is indistinguishable from a non-modular random graph. This assumption allowed the development of networks with defined block structure and it was found that this structure places strong constraints on the level of positive assortativity possible.

Whilst capable of generating rich, complex structure, including a wide range of spectral properties, fitting CiGRAM requires computationally sophisticated methods of estimation. In Chapter 5, approaches to fitting real world networks with particle swarm optimisation were evaluated. Fitting the eigenvalue distributions of normalised Laplacian matrices was found to be limited in terms of the computational feasibility and the quality of fit for other graph summary statistics such as the desired degree distributions and clustering, and assortativity coefficients.

The evaluation of fitting the degree distributions, assortativity and clustering coefficients directly, was shown to be a reasonable alternative approach. This method scales to larger networks and has the ability to fit the desired target summary statistics to a good degree of accuracy. The selected model parameters, however, fail to fit other summary statistics such as shortest paths and centrality for target graphs, highlighting that more complete distance metrics are required. However, the best fit models still provide a strong basis for the evaluation of module detection algorithms in a practical context.

The penultimate chapter of this thesis evaluated algorithms in different contexts. By modelling degree assortativity, a property ignored in many other ground-truth benchmark networks, it was found that modularity maximisation and Infomap based module detection algorithms perform significantly worse in the presence of positive degree assortativity. This is likely due to the assumptions in statistical and information theoretic methods that include specific null models and provides potential insight into the development of new algorithms.

The evaluation of algorithms in a practical context was then performed. This

methodology forms the basis of algorithm selection for empirical datasets and it was found that many module extraction methods perform poorly under these test conditions. These results disagree with previous structural benchmarks [159,196], which find algorithms perform well with high levels of mixing between communities. The performance, however, is similar to results in social networks that include meta-data for real communities [194]. Here, it was found that many algorithms were unable to detect meta-data communities, indicating that significant improvements in algorithms need to be made in more practical contexts.

7.2 Conclusions

Several specific conclusions can be drawn from this project. One of the most important conclusions of this study is that the level of agreement between different module extraction methods is very low. This lack of agreement makes it difficult to justify the selection on any algorithms.

The development of a model for modular networks has given a definition of community structure that a bottom level module is a subgraph that is indistinguishable from a non-modular random graph. This definition makes no assumptions about the detectability of modules but allows the modelling of graphs with highly modular structure. This modelling approach clearly demonstrates that generating networks with modular structure results in significantly higher clustering coefficients (transitivity) than one would expect in non-modular configurations. Furthermore, the geometric approach used by CiGRAM allows us to draw several interesting conclusions. The heterogeneity of the degree distribution and the assortative configurations can be thought of in geometric terms. It is important to note, however, that the geometry modelled by these approaches is not a "real" geometry underlying any datasets. Unlike the work of Papadopoulos et al. [126], no statements about this space are made. Indeed, unless one has conclusive evidence of any geometric space it is always possible to argue that another hypothetical model is a better fit for any underlying graph.

The assortativity modelled by CiGRAM also suggests that positive degree

assortativity requires sparse graphs. Whilst this fact is not formally proved in this thesis, the evidence suggests that dense configurations and certain modular configurations of networks prevent positive assortativity from forming. This is despite very high values for the parameters that increase the propensity for nodes of the same degree to form edges.

The analysis of the performance of algorithms on modular structure lead to some interesting findings. Firstly, the performance of the infomap and modularity based algorithms appears to be impacted by assortativity. This implies that, for extremely assortative networks users should consider alternative algorithms, such as those tested here. Furthermore, when evaluated on realistic topology, many of the algorithms failed to uncover any ground-truth topology. This disagrees with some of the literature for existing benchmarks, that ranked many algorithms very highly. This result did not appear to be caused by the level of overlap and mixing between communities. The implication we can draw from this is that richer topology has a strong impact upon algorithm performance.

7.3 Limitations

The approach to generating complex networks with CiGRAM has several notable limitations. Firstly, the model is only capable of generating undirected and unweighted graphs, this means that it is simply unable to form a model for many of the datasets researchers would like to evaluate. The approach taken here also includes no notion of hierarchical modular structure, a feature that is explored in several recent studies.

Perhaps the biggest limitation of CiGRAM, however, is the difficulty of selecting the correct parameters for a given network. Many of these parameters interact with each other, requiring non-trivial heuristics to control the graph structure in order to fit desired topological features. This means that fitting larger datasets is more complicated, potentially being extremely time consuming with the methods explored as part of this thesis. The generation time for networks is also a strong limitation of the CiGRAM algorithm, for very large networks the methods proposed here would quickly become intractable.

In Chapter 6, the evaluation of module extraction algorithms was conducted in the presence of fixed levels of degree assortativity. Here it was found that CiGRAM generated networks with degree assortativity coefficients across a wide range of values. This limited the tests as a re-sampling approach was applied to ensure that the correct levels of degree assortativity were met. This limitation may be apparent for more topological properties and is an inherent issue to all probabilistic modelling approaches.

Furthermore, when evaluating the performance of algorithms on best fit models, it is important to point out that none of these can be said to be exact matches to the real world topologies. Given that, for each network, only a single topological value is fit with the model, the evaluation of model fit is extremely challenging. Moreover, the summary statistics method does not capture all topological properties and interpretation of the results must be taken with great care.

7.4 Contributions

This section summarises the core major and minor contributions to knowledge presented in this thesis.

7.4.1 Major contributions

Analysis of topological module extraction in coexpression networks

Chapter 3 presented a comparative analysis of module extraction algorithms in the context of plant correlation of expression networks. This work formally tests the performance of algorithms against one another. The results showed that there is little formal agreement between algorithms, making it difficult to justify the selection of results. Validating results in terms of gene knock-out experiments and gene ontology provided some evaluation of the performance, showing that many algorithms detect statistically meaningful clusters. The analysis of phylogenetic data also assisted in this regard, aiding hypothesis generation about the function of gene clusters given their evolutionarily conserved nature. However, there are few methods to validate the underlying modules;

this limits the potential for characterising unknown genes.

A definition of modular structure

One of the most fundamental contributions of this thesis, beyond the construction of modular random graphs, is a definition of what a module is. Under this definition, the argument is that there is no distinction between a random non-modular network and a module. This definition does take into account the detectability of modules, but it is a clear working definition of what a module is. This has implications for many community extraction approaches that either implicitly or explicitly include null models for community detection.

A realistic synthetic model for networks

The most significant contribution of this thesis is CiGRAM, which provides a model for generating modular complex networks with a fixed edge density. This approach is similar to both geometric [126] and stochastic block models [130] in some respects. However, there are several core distinctions that need to be highlighted. The use of wrapped Gaussian distributions to modify the heterogeneity of graphs allows a flexible approach to generating degree distributions. This approach makes minimal assumptions about any underlying process that generates networks and does not require fixed power laws to generate heavy tailed degree distributions. Furthermore, the generation of assortativity through geometric variables provides an entirely novel approach to generating this form of structure. This modelling approach allowed the discovery of a potential fact that assortative structure may necessarily require a level of sparsity and mixing between modules. Through parameter optimisation with particle swarm optimisation in Chapter 6, it was shown that CiGRAM is capable of generating the rich and diverse topological structure found in empirical data.

Methodology for module extraction algorithm selection

Evaluation of algorithms in a practical context is a difficult challenge given that minimal information about the true modular structure of biological networks is known. Furthermore, topology based module extraction uses no additional

information and attempts to classify nodes according to topological structure alone. As CiGRAM generates networks with a known modular structure it can be used to evaluate algorithms. Using particle swarm optimisation, the parameters of this model were tuned to closely match the topology of real world networks. This presents an approach that researchers can use to aid algorithm selection where no data about the true community structure is known *a priori*. In Chapter 6, this approach found that algorithms appear to perform poorly in the context of these models. This may, in part, be a result of the inability of CiGRAM to fully represent the topology of real world networks. However, as models improve, this approach can still be used to evaluate algorithms in domain specific contexts.

Impact of assortativity on module detection

Another contribution of this work is the discovery that assortativity has an impact upon the performance of some module detection algorithms. Specifically, the Infomap and modularity maximisation approaches performed significantly worse in the presence of assortativity structure, whilst algorithms such as label propagation and OSLOM did not appear to be impacted. Assortativity is an important topological property that does not readily occur by chance alone; null models that are used in these algorithms could be used to improve the performance in this context.

7.4.2 Minor contributions

Web visualisation tool

Appendix A also presented a web visualisation tool for large-scale correlation of expression datasets. This approach aids bioscientists by providing a complementary user interface to publications for gene expression experiments. By providing the material in a web application, users can explore experimental results without being required to download large datasets. Furthermore, the application only displays a subset of edges at any given time, allowing the visualisation to run on slower systems such as mobile devices.

A novel distance metric for graph spectra

In Chapter 5, the use of the cumulative spectral distance to compare networks was presented. This enabled the utilisation of the Kolmogorov-Smirnov distance in the context of network distances. In terms of comparing graph spectra, this approach allows distance between graphs that are of different sizes. As all normalised Laplacian eigenvalues are, necessarily, in the range $[0, 2]$, this cannot be achieved with euclidean distance metrics. Furthermore, approaches that use the distributions of graph spectra require the use of either histograms or Gaussian kernel estimates over mass functions of the graph spectra. Given that cumulative spectra are defined over a continuous range, the KS distance does not suffer from this limitation.

7.5 Future work

The following section reviews some of the potential directions for future work.

Improvements for network fit A core limitation of the work presented in this thesis is measuring the ability with which CiGRAM can fit real world networks. There are two possible approaches that could be used to achieve this goal; the use of a likelihood function, as found in stochastic block models [130], or the development of an improved distance metric for graphs. The likelihood approach is limited by the fixed density nature of CiGRAM; the sampling without replacement process means there is no closed form solution to the probability of a given graph and one must iterate over all possible permutations for generating a network. One way around this could be the development of a pseudo-likelihood function [197] that circumvents this issue, perhaps removing the formal condition of fixed density.

In terms of distance metrics, the graph edit distance [180] was ruled out for reasons of computational complexity. Computing the minimum number of edge rewirings required to generate an isomorphic graph will never be trivial to compute, though some form of estimation may be feasible. Alternatively, the use of small graph sub-structures, *motifs*, has been used to quantify the difference between networks [198].

Use of other probability distributions in CiGRAM The use of

wrapped normal probability density functions in CiGRAM proved to be a flexible approach to the generation of heterogeneity. However, in the case of communities, this approach proved to be very unpredictable as K is typically relatively small. Furthermore, these distributions were not formally evaluated against other wrapped distributions such as the wrapped Cauchy or the more general family of wrapped exponential distributions [174]. More research into how these approaches can be used to estimate graph topology would likely improve CiGRAM, possibly even improving the fit to real world networks.

Another approach could be to form a Markov Chain Monte Carlo algorithm where to uncover the best fit distribution of latent variables and fitness functions. Such an approach, however, would need more formal approaches at evaluating the fit to the target distributions via maximum likelihood estimation rather than the Kolmogorov-Smirnov distance used in this thesis.

Further integration of modules into web visualisation The web visualisation presented in Chapter 3 displays high-throughput large scale biological data in a convenient manner that accompanies publications. The use of module extraction approaches aids the visualisation of networks but this could be improved with further work. Allowing users to upload their own datasets and selecting a module detection algorithm with high confidence, using CiGRAM or other benchmark graphs, is an approach that could also be useful to bioscientists. Furthermore, a more formal approach to using module extraction could be developed by integrating multiple data sources, and providing users with related genes based on key word queries combined with modular structure, offering an interesting aid to hypothesis generation.

Generation of directed and weighted networks As a model of undirected complex biological networks, CiGRAM appears to perform well. However, causal links in metabolic reactions and genetic regulatory networks are vital to modelling the behaviour of systems [2]. This also relates strongly to the less well researched areas of module detection in complex biological networks. Furthermore, weighted links need to be considered when evaluating the strengths of interactions and connected components. The implementation of directed links may prove simple, given CiGRAMs two step connection process. However, significant work needs to be conducted into the modelling of in and out degrees.

The distribution of weights is also an issue that requires further analysis, but it may be achieved through replacing a sampling without replacement procedure with a sampling with replacement procedure.

Modelling hierarchical systems Many biological systems are thought to be hierarchical in nature [52, 199] and module detection methods such as Infomap [96] and OSLOM [67] attempt to uncover hierarchical organisation. Whilst this is a topic not discussed in this thesis, the assumptions of CiGRAM actually lend themselves to hierarchical construction. If a bottom level module is defined as a non-modular random graph, successive levels of hierarchical organisation can be defined allowing the generation of hierarchical structure. A problem with this method, however, is that it is difficult to isolate specific topological summary statistics that indicate the presence of hierarchy. Further research into this would present a significant contribution to knowledge.

Further analysis of assortativity Results in this thesis appear to indicate that assortative structure is often absent in dense random graphs and that modular structure strongly influences the resulting degree assortativity of real networks. The work presented here is not fully conclusive. Further analytical and statistical evidence should be provided to test to see if, as with scale-free topology [10], assortative topology requires sparse graphs. If so, this would have strong implications for any latent modular structure.

Appendix A

Web visualisation tool

The interactive tool developed for this work is available at `http://netvis.ico2s.org/endonet/` and `http://netvis.ico2s.org/radnet/`. This visualisation was presented as part of the work for Dekkers *et al.* [147] and provides an interactive component to the paper, improving the visualisation created for SeedNet [43] and SCopNet [168]. An additional tool is to be included for FruitNet [148] upon publication. Figure A.1 demonstrates the interfaces of the RadNet tool. Genes are annotated with the TAIR [135] and SolGenomics [200] data for *Arabidopsis* and Tomato data, respectively. The remainder of the section discusses the implementation and functionality of the tool.

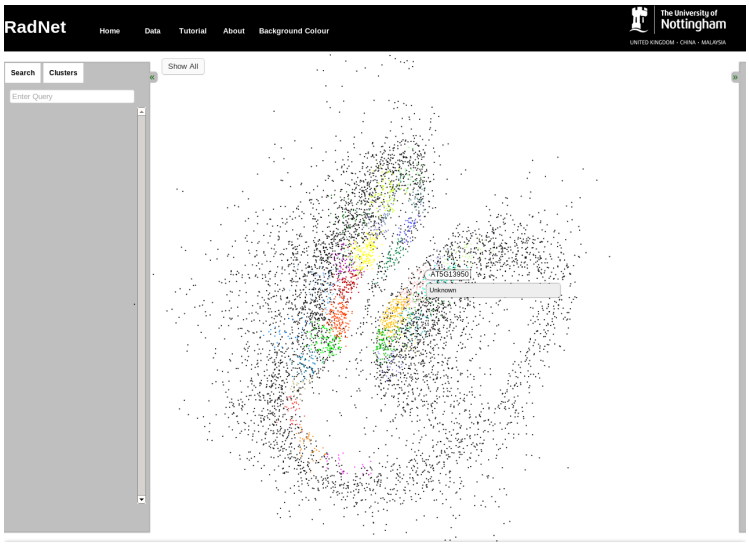


Figure A.1: Web based visualisation of RadNet network.

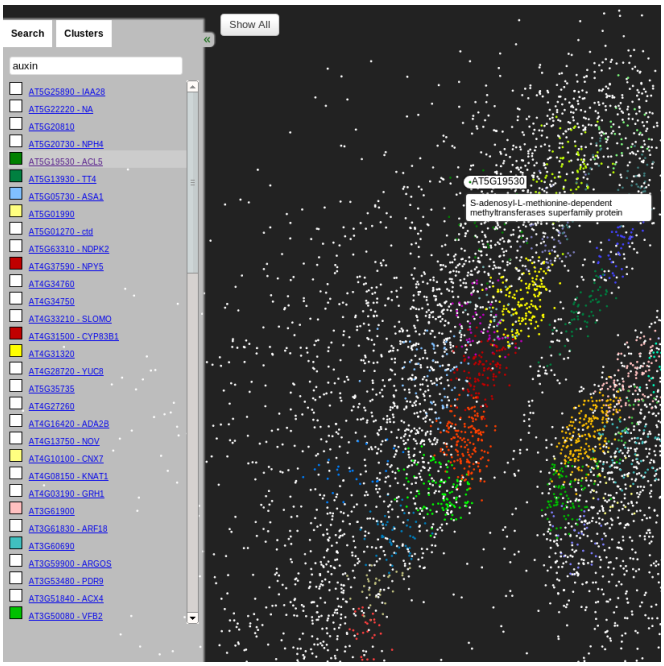
In a similar vein to other visualisation software such as Cytoscape Web [201] and ONDEX Web [202], the platform is designed to be web accessible. The

implementation of the tool is designed to be a companion to research papers and to display large networks. This functionality is achieved by only displaying edges between selected nodes and their neighbours, this allows the visualisation to run on significantly less powerful systems. Furthermore, the tool is written in Javascript making use of the HTML5 canvas element and does not require any additional plug-ins or software. The links for all nodes are fully exportable in Javascript Object Notation (JSON) and CSV format, allowing external APIs to connect to the tool through queries.

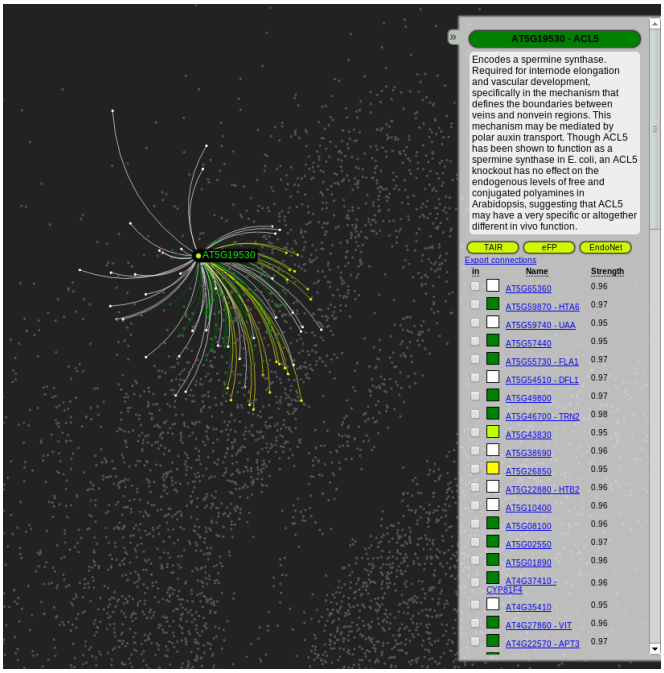
The tool also includes keyword search functionality and allows the highlighting of detected modules. In SeedNet the modules include genes associated with Up and Down regulation during germination. In RadNet and EndoNet the detected modules relate to detected clusters inline with the timecourse of the experiment detected with the MCODE [136] clustering algorithm included in the article [147] [1]. Figure A.2 demonstrates the interface for the search and gene view features.

The layout in the FruitNet web visualisation is based on the underlying modular structure of the network and uses the CVIS layout included in the OSLOM algorithm [67]. This means that the clusters related to co-expressed genes, rather than force directed layout modified for aesthetic visualisation. This is shown in Figure A.3.

[1] The work with MCODE [136] was conducted by collaborators in [147] and is not part of this thesis



(a) Search View



(b) Gene View

Figure A.2: Search and gene view interfaces to the Network web visualisation tool. The search tool includes keyword highlighting within the network from associated gene annotations.

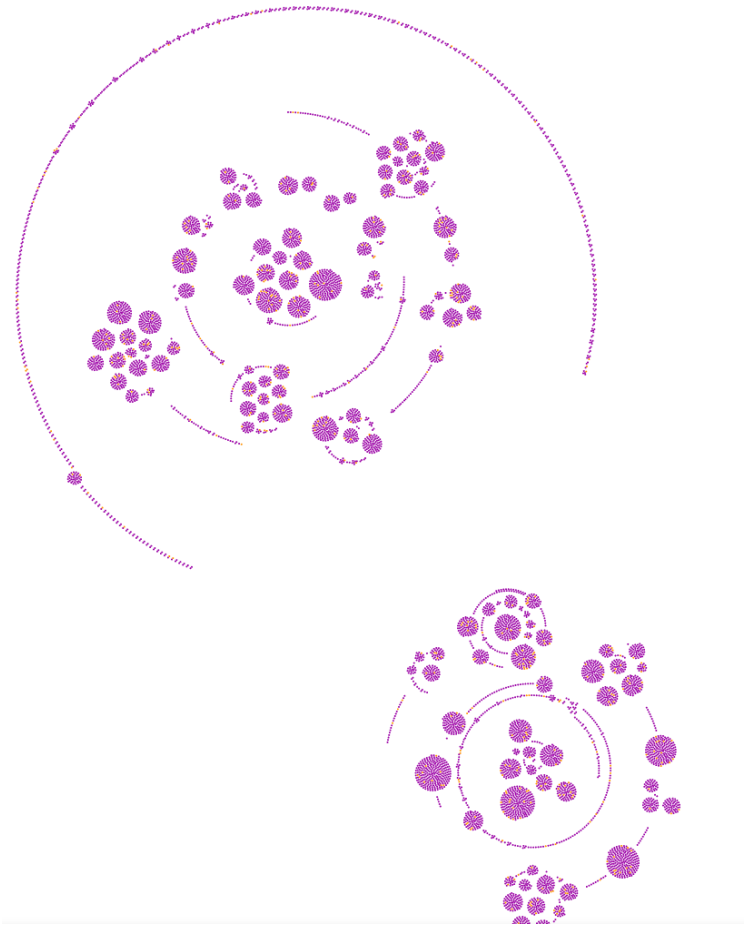


Figure A.3: Cluster based visualisation of FruitNet in web based network visualisation.

Appendix B

Model parameter selection supplement

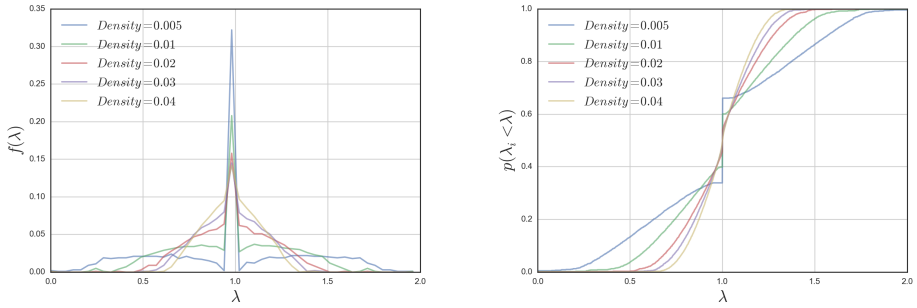
B.1 CiGRAM and graph spectra

In this section, we observe spectral properties of the best fit networks in the form of the eigenvalues of the normalised Laplacian matrix. The normalised Laplacian of a graph is defined in Equation 2.10 and has several interesting properties that make it appealing from the perspective of comparative analysis of graphs. Because all the eigenvalues are real and necessarily fall in the range $[0, 2)$ [181], one can compare graphs across vastly different scales. This section observes the plots of the eigenvalue distributions.

B.1.1 Parameter influence on spectra

The following accounts how the parameters of CiGRAM influence the Normalised Laplacian structure of graphs. The graph spectra appears to be heavily influenced by the density of generated graphs. Figure B.1 shows how increasing the density of a graph changes the resulting spectral distribution. With all other CiGRAM parameters changing, the peaks and general shapes of the distribution appear to be heavily dependent on the resulting density of the generated networks. As the density increases the peaks of the graph become less pronounced with less spread over the eigenvalues of the distribution.

At a fixed density, the peaks of the distribution appear to be determined by



(a) Spectral distribution *(b) Cumulative spectral distribution*

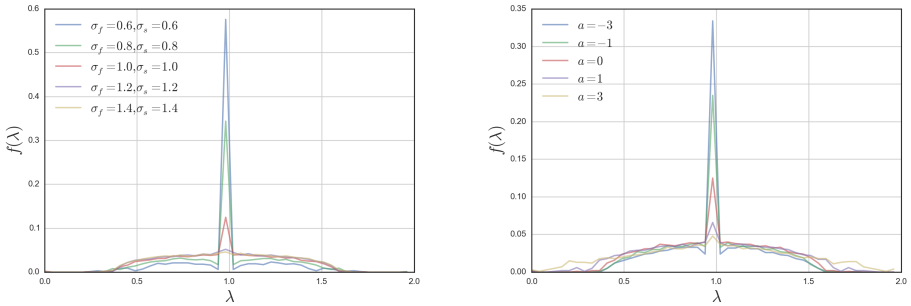
Figure B.1: Influence of density on the resulting normalised Laplacian spectra. Networks were generated with CiGRAM 1000 nodes and parameters $k = 1$, $\sigma_f = 1.0$, $\sigma_s = 1.0$, $a = 0$.

the degree distribution and assortativity of the graphs. Figure B.2 (a) shows that the peak of the distribution appears to be strongly influenced by the node position and scoring parameters σ_f and σ_s. However, the assortativity parameter a also appears to have some degree of impact on the peaks of the distribution, implying competition between these parameters, as shown previously in Chapter 4. The cumulative distributions in Figure B.2 (c) and (d) show that the assortativity parameter also influences the spread of the distributions in a manner that is not found only by modifying σ_f and σ_s.

Figure B.3 (a) shows that K appears to add a large degree of noise to the resulting spectral distributions. The spectra appears far less well behaved than the single community spectral distributions shown in Figure B.2, with less clearly defined peaks. The cumulative spectral distribution in Figure B.3 (a) shows that eigenvalues below $\lambda < 0.4$ appear to be more numerous with increased values of k. Figure B.4 e_k and p_o show that, whilst these parameters have some influence on the spectra, it is far less visible, with p_o having very little detectable change to the distribution of eigenvalues.

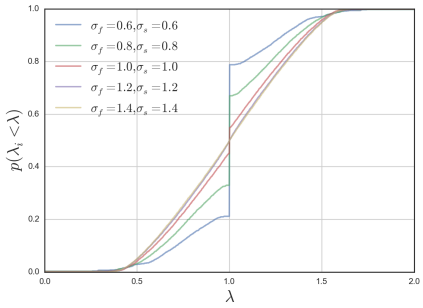
B.2 Additional fitting results

The remainder of this appendix contains tables and figures relating to the fits in Chapter 5. Table B.1 shows the best fit CiGRAM parameters of the networks under the different model conditions. The selected parameters fall over a wide

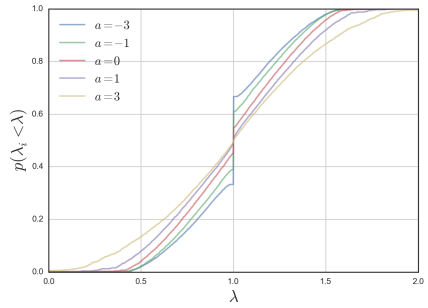


(a) Spectral dist σ

(b) Spectral dist a



(c) Cumulative dist σ



(d) Cumulative dist a

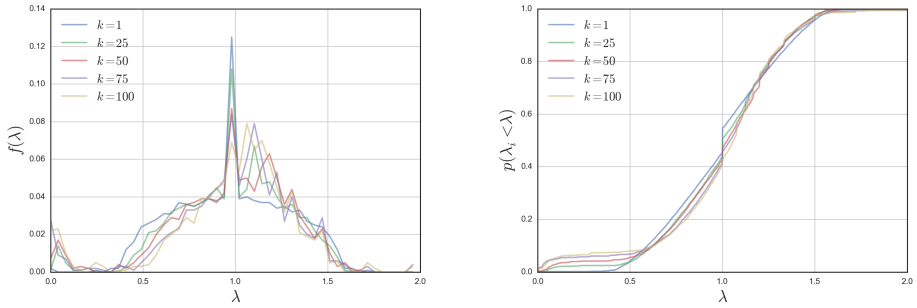
Figure B.2: Spectral and cumulative spectral distributions for varying levels of σ and a parameters. Networks were generated with 1000 nodes and fixed density of 0.01*. Where σ_f and σ_s vary, $a = 0$. Where a varies, $\sigma_f = 1.0$ and $\sigma_s = 1.0$.*

range for each network indicating that CiGRAM is capable of generating similar topological fits. This also indicates the difficulty the optimisation process has as it indicates there are many local optima.

Figures B.5 to B.7 and Table B.2 show the topological properties of the model networks, highlighting the ability and inability of the fitting procedure to accurately represent topology not directly measured in the fitness functions of equation 5.11 and 5.12.

Figure B.5 measures the mean shortest path length (SPL) of the networks under study described in equation 2.3. SPL is an interesting topological property in these circumstances as the results show that matching the degree distribution and clustering coefficients of empirical data is not sufficient to generate real world topology.

Figure B.6 shows the central point dominance (CPD) of networks, described in equation 2.6, which relates to the betweenness centrality of nodes. CPD is closely related to SPL in the sense that Equation 2.6 is based on betweenness



(a) Spectral distributions *(b) Cumulative spectral distributions*

Figure B.3: Spectral and cumulative spectral distributions varying K. Networks were generated with CiGRAM 1000 nodes and parameters $\sigma_f = 1.0$, $\sigma_s = 1.0$, $a = 0$, $e_k = 0.1$, $p_o = 0.0$ $\tilde{\sigma}_s = 1$ $\tilde{\sigma}_f = 1$.

centrality, which counts the number of shortest paths through a node.

Figure B.7 shows the modularity (see equation 2.11) of the best fit models. The modularity of networks is used to optimise parititions to uncover community structure, but it can also be considered an indication of the block structure of networks. These results indicate that matching simple topological properties is not sufficient to accurately model modular structure.

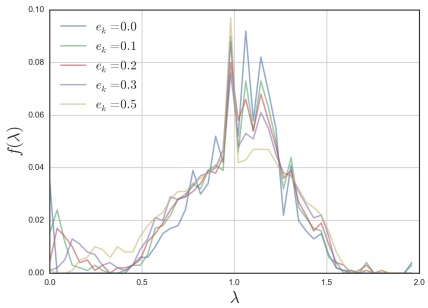
As with other topological properties, the spectral distributions for the networks vary over a wide range. Whilst these results indicate that the fitting procedure was unable to find a good match for graph spectra, Section 5.4 revealed that fitting spectra, alone, does not guarantee fits for other topological properties. Figure B.8 and B.9 show distribution of the eigenvalues of the Normalised Laplacian matrix.

Network	experiment	σ_f	σ_s	a	K	$\tilde{\sigma}_f$	$\tilde{\sigma}_s$	e_k	p_o	Fit
Yeast PPI	Single K	1.427	0.899	-0.176	-	-	-	-	-	0.019
	Fixed K	1.6	0.879	-0.055	43	0.337	1.146	0.329	0.05	0.02
	Free params	1.6	0.83	0.0	228	1.6	0.926	0.5	0.497	0.026
	Low overlap	1.6	0.85	-0.184	228	0.809	0.534	0.1	0.01	0.044
	High Overlap	1.361	0.845	0.0	47	1.48	0.919	0.25	0.1	0.031
Arabidopsis PPI	Single K	0.96	0.884	-1.033	-	-	-	-	-	0.079
	Fixed K	0.841	0.853	-1.0	41	1.024	0.407	0.185	0.05	0.03
	Free params	0.1	0.352	-1.296	186	1.54	1.596	0.338	0.39	0.113
	Low overlap	1.6	0.484	-6.0	162	0.945	1.102	0.1	0.01	0.167
	High Overlap	1.6	0.6	-4.885	78	1.528	1.6	0.25	0.1	0.054
C Elegans Metabolic	Single K	0.388	0.84	-4.991	-	-	-	-	-	0.064
	Fixed K	0.941	1.3	-4.746	8	0.467	0.1	0.351	0.05	0.069
	Free params	0.202	0.512	0.0	20	2.093	0.449	0.039	0.162	0.217
	Low overlap	0.532	0.696	-5.0	13	1.326	0.927	0.1	0.01	0.171
	High Overlap	0.345	0.634	-4.97	13	0.827	1.406	0.25	0.1	0.261
E coli Metabolic	Single K	0.961	1.059	6.0	-	-	-	-	-	0.031
	Fixed K	1.022	1.024	6.0	10	0.25	0.296	0.151	0.05	0.128
	Free params	1.586	0.876	5.975	24	1.033	0.756	0.01	0.01	0.086
	Low overlap	2.105	0.988	6.0	25	0.1	0.368	0.1	0.01	0.14
	High Overlap	0.86	0.999	5.773	1	1.276	0.1	0.25	0.1	0.159
SeedNet	Single K	1.242	0.81	0.587	-	-	-	-	-	0.152
	Fixed K	1.6	0.808	1.49	28	1.57	0.487	0.075	0.05	0.126
	Free params	1.6	0.793	0.923	30	1.6	0.565	0.022	0.01	0.192
	Low overlap	1.599	0.802	1.002	356	1.517	0.1	0.1	0.01	0.226
	High Overlap	1.6	0.753	0.0	604	1.331	0.1	0.25	0.1	0.25
Open Flights	Single K	0.761	0.756	1.124	-	-	-	-	-	0.031
	Fixed K	0.786	0.758	1.497	24	1.59	0.466	0.163	0.05	0.049
	Free params	2.2	0.566	0.299	132	1.942	2.2	0.5	0.01	0.077
	Low overlap	2.2	0.407	3.0	290	0.909	0.579	0.1	0.01	0.154
	High Overlap	1.425	0.597	0.921	267	0.614	0.738	0.25	0.1	0.053
US Power Grid	Single K	0.455	1.121	0.145	-	-	-	-	-	0.024
	Fixed K	0.13	0.705	0.713	45	0.487	1.545	0.047	0.05	0.018
	Free params	0.214	0.829	0.128	182	1.582	1.588	0.01	0.04	0.006
	Low overlap	0.967	1.6	-0.45	294	1.597	1.063	0.1	0.01	0.036
	High Overlap	0.597	1.432	-0.128	420	1.512	1.213	0.25	0.1	0.015
PGP	Single K	1.059	0.829	2.21	-	-	-	-	-	0.029
	Fixed K	0.716	0.74	3.339	198	1.136	0.686	0.085	0.05	0.012
	Free params	1.031	0.596	4.651	306	0.183	0.532	0.035	0.133	0.031
	Low overlap	0.994	0.694	2.606	389	0.476	0.785	0.1	0.01	0.023
	High Overlap	1.521	0.643	1.0	1021	0.387	0.786	0.25	0.1	0.12
Hamster	Single K	0.645	0.824	0.903	-	-	-	-	-	0.036
	Fixed K	0.6	0.779	1.673	13	0.1	0.535	0.339	0.05	0.039
	Free params	0.544	0.767	1.271	51	1.094	0.1	0.01	0.015	0.109
	Low overlap	0.675	0.835	1.018	147	0.747	0.197	0.1	0.01	0.081
	High Overlap	0.798	0.863	0.889	7	0.1	0.581	0.25	0.1	0.075

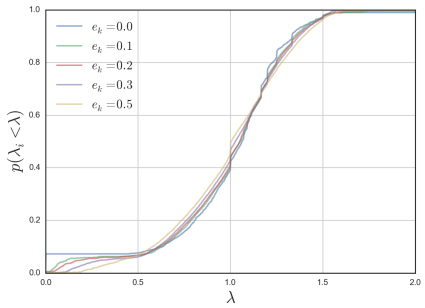
Table B.1: CiGRAM best fit parameters discovered with particle swarm optimisation. These parameters are more fully described in Chapter 4 Table 4.1

Experiment	Network	$S\hat{P}L$	p	$C\hat{P}D$	p	$\hat{Q}$	p	D_{js}	D_{ks}	D_e
Single K	Yeast PPI	4.031	0.0	0.044	0.0	0.401	0.0	0.155	0.042	0.058
	Arabidopsis PPI	4.149	0.0	0.097	0.396	0.45	0.0	0.19	0.049	0.055
	C Elegans Metabolic	2.397	0.08	0.411	0.479	0.268	0.0	0.337	0.099	0.103
	E coli Metabolic	4.994	0.004	0.09	0.008	0.556	0.0	0.193	0.054	0.054
	SeedNet	2.644	0.0	0.013	0.0	0.115	0.0	0.249	0.076	0.112
	Open Flights	3.395	0.0	0.06	0.039	0.246	0.0	0.166	0.049	0.058
	US Power Grid	8.572	0.0	0.05	0.0	0.739	0.0	0.073	0.021	0.029
	PGP	5.488	0.0	0.062	0.0	0.497	0.0	0.153	0.053	0.087
	Hamster	3.009	0.0	0.076	0.395	0.225	0.0	0.292	0.137	0.083
Fixed K	Yeast PPI	4.171	0.0	0.052	0.173	0.598	0.856	0.148	0.038	0.057
	Arabidopsis PPI	4.0	0.0	0.13	0.678	0.465	0.0	0.182	0.047	0.055
	C Elegans Metabolic	2.416	0.052	0.431	0.512	0.267	0.0	0.308	0.072	0.09
	E coli Metabolic	4.991	0.047	0.159	0.594	0.554	0.0	0.185	0.056	0.054
	SeedNet	2.805	0.0	0.018	0.0	0.543	0.463	0.163	0.051	0.081
	Open Flights	3.519	0.0	0.072	0.281	0.436	0.015	0.154	0.039	0.049
	US Power Grid	10.522	0.0	0.058	0.0	0.879	0.0	0.062	0.02	0.027
	PGP	5.72	0.0	0.289	1.0	0.817	0.0	0.142	0.052	0.087
	Hamster	3.028	0.0	0.101	0.694	0.5	0.979	0.293	0.146	0.081
Free params	Yeast PPI	4.096	0.0	0.046	0.025	0.516	0.0	0.135	0.041	0.053
	Arabidopsis PPI	4.004	0.0	0.181	0.836	0.538	0.0	0.17	0.039	0.052
	C Elegans Metabolic	2.8	0.778	0.338	0.274	0.594	0.926	0.301	0.08	0.087
	E coli Metabolic	2.709	0.556	0.153	0.568	0.788	0.968	0.224	0.075	0.066
	SeedNet	2.863	0.0	0.026	0.0	0.703	0.963	0.253	0.095	0.12
	Open Flights	3.463	0.0	0.09	0.549	0.467	0.0	0.191	0.066	0.094
	US Power Grid	12.938	0.973	0.121	0.0	0.971	1.0	0.044	0.02	0.016
	PGP	5.802	0.0	0.3	1.0	0.849	0.01	0.093	0.037	0.055
	Hamster	2.998	0.0	0.118	0.754	0.245	0.0	0.286	0.134	0.083
Low overlap	Yeast PPI	4.414	0.711	0.053	0.1	0.707	0.976	0.135	0.037	0.047
	Arabidopsis PPI	4.35	0.0	0.394	1.0	0.799	1.0	0.236	0.13	0.16
	C Elegans Metabolic	2.715	0.665	0.336	0.235	0.543	0.938	0.44	0.189	0.143
	E coli Metabolic	5.023	0.222	0.114	0.213	0.725	0.879	0.227	0.065	0.059
	SeedNet	2.776	0.0	0.019	0.0	0.599	0.667	0.257	0.092	0.12
	Open Flights	3.963	0.112	0.444	1.0	0.457	0.0	0.207	0.099	0.094
	US Power Grid	13.077	0.0	0.063	0.0	0.935	0.426	0.051	0.024	0.02
	PGP	5.51	0.0	0.323	1.0	0.871	0.036	0.104	0.03	0.034
	Hamster	3.026	0.0	0.107	0.623	0.275	0.026	0.294	0.142	0.084
High Overlap	Yeast PPI	4.178	0.0	0.049	0.022	0.671	1.0	0.125	0.035	0.046
	Arabidopsis PPI	4.232	0.0	0.311	0.994	0.649	0.0	0.243	0.13	0.158
	C Elegans Metabolic	2.716	0.718	0.275	0.063	0.515	0.99	0.366	0.131	0.118
	E coli Metabolic	5.011	0.002	0.091	0.016	0.533	0.0	0.19	0.06	0.058
	SeedNet	2.628	0.0	0.013	0.0	0.504	0.306	0.328	0.123	0.143
	Open Flights	3.558	0.0	0.118	0.923	0.591	0.009	0.163	0.056	0.075
	US Power Grid	9.915	0.0	0.048	0.0	0.851	0.0	0.03	0.021	0.015
	PGP	4.998	0.0	0.169	0.859	0.714	0.0	0.178	0.055	0.083
	Hamster	3.087	0.0	0.071	0.346	0.481	0.969	0.297	0.139	0.084

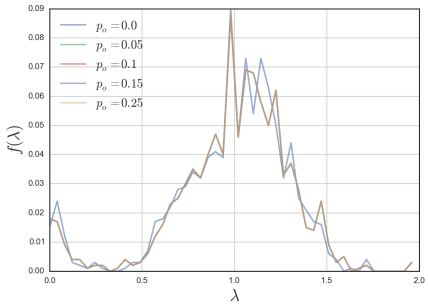
Table B.2: Topological results for best fit models. Results shown are the mean of 100 samples with the best fit CiGRAM parameters.



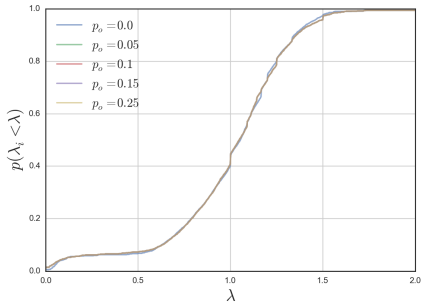
(a) e_k spectral distributions



(b) e_k cumulative spectral distributions

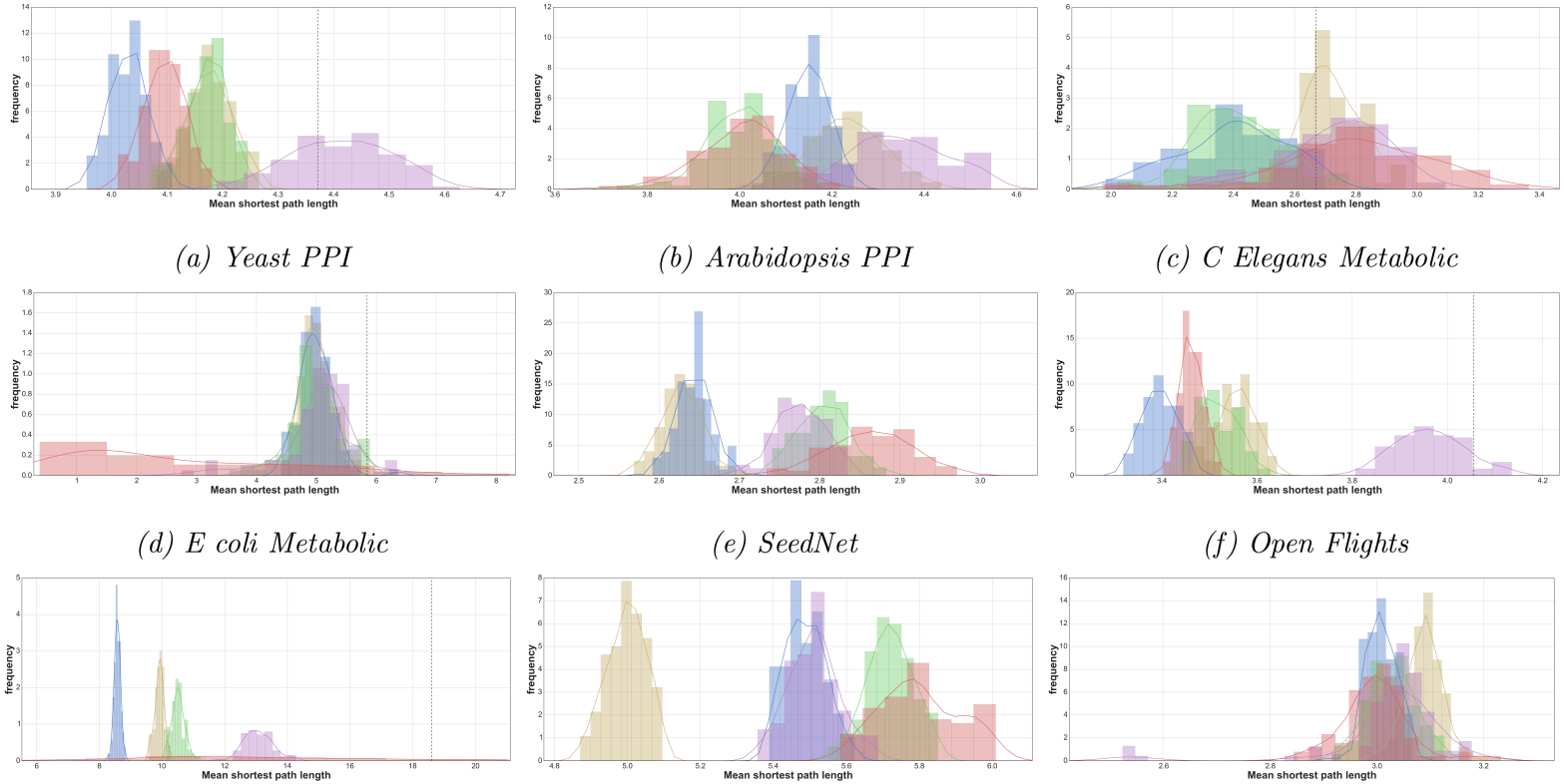


(c) p_o spectral distributions



(d) p_o cumulative spectral distributions

Figure B.4: Spectral and cumulative spectral distributions varying e_k and p_o. Networks were generated with CiGRAM 1000 nodes and parameters $k = 80$, $\sigma_f = 1.0$, $\sigma_s = 1.0$, $a = 0$, $\tilde{\sigma}_s = 1$ $\tilde{\sigma}_f = 1$.

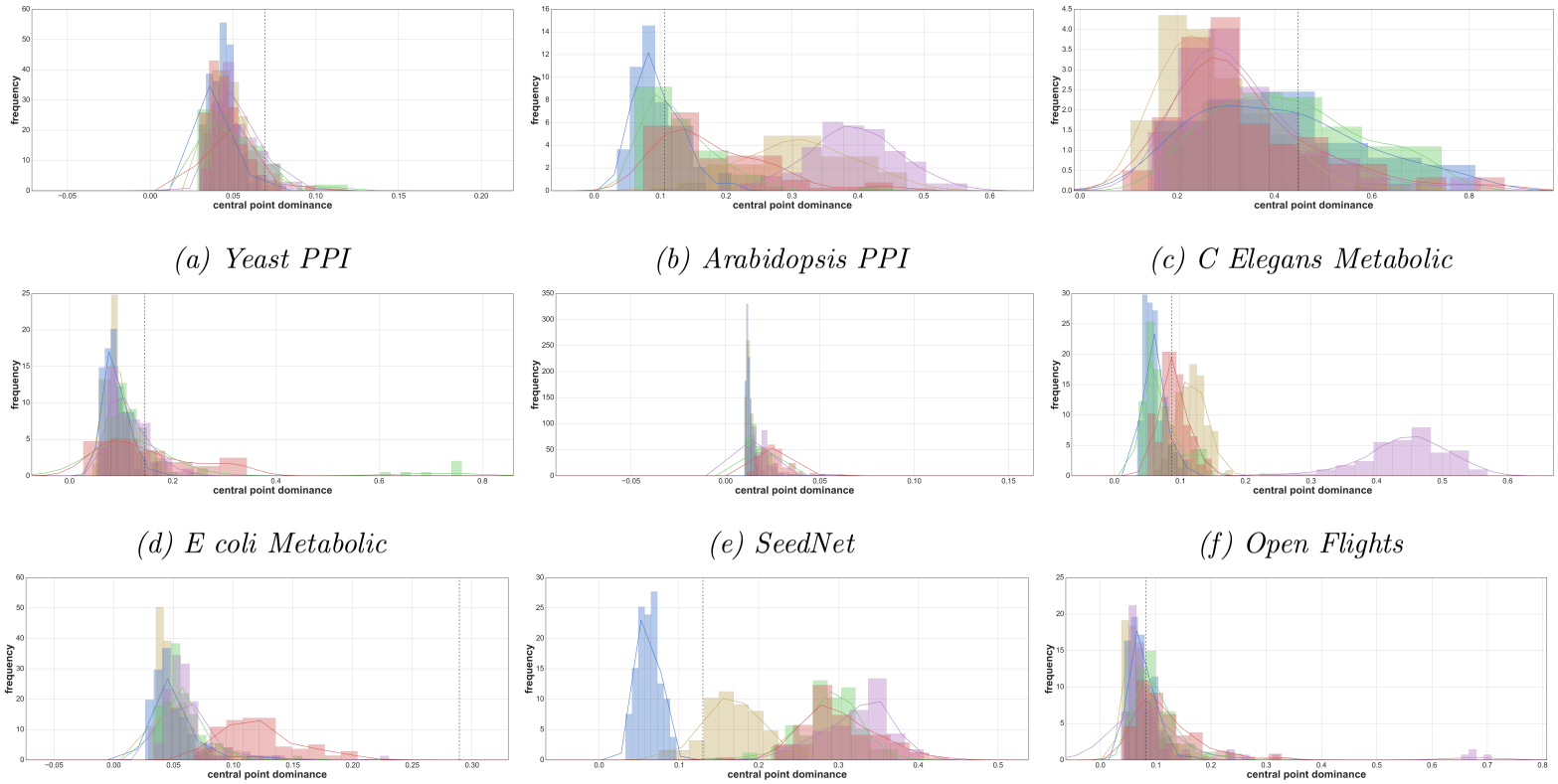


(a) Yeast PPI *(b) Arabidopsis PPI* *(c) C Elegans Metabolic*

(d) E coli Metabolic *(e) SeedNet* *(f) Open Flights*

(g) US Power Grid *(h) PGP* *(i) Hamster*

Figure B.5: Distribution of mean shortest path length for best fit models. Histograms of 100 samples with kernel density estimates are shown. Colours indicate model fit for real world graphs (black dashes), single K (blue), fixed K (green), free parameters (red), low overlap (purple) and high overlap (yellow). Where dashes are not present, this is due to high levels of model inaccuracy.

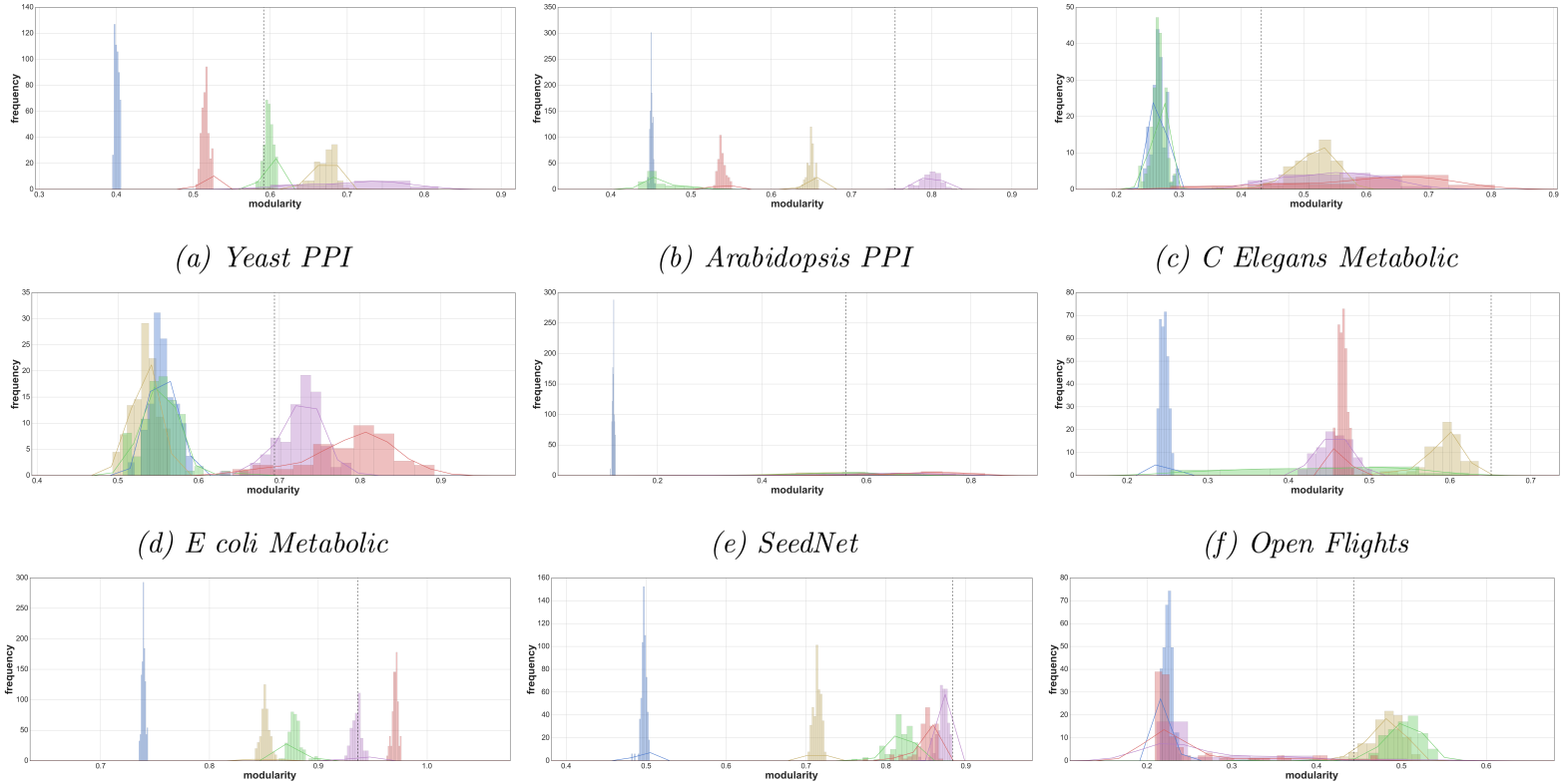


(a) Yeast PPI *(b) Arabidopsis PPI* *(c) C Elegans Metabolic*

(d) E coli Metabolic *(e) SeedNet* *(f) Open Flights*

(g) US Power Grid *(h) PGP* *(i) Hamster*

Figure B.6: Distribution of central point dominance for best fit models. Histograms of 100 samples with kernel density estimates are shown.Colours indicate model fit for real world graphs (black dashes), single K (blue), fixed K (green), free parameters (red), low overlap (purple) and high overlap (yellow). Where dashes are not present, this is due to high levels of model inaccuracy.

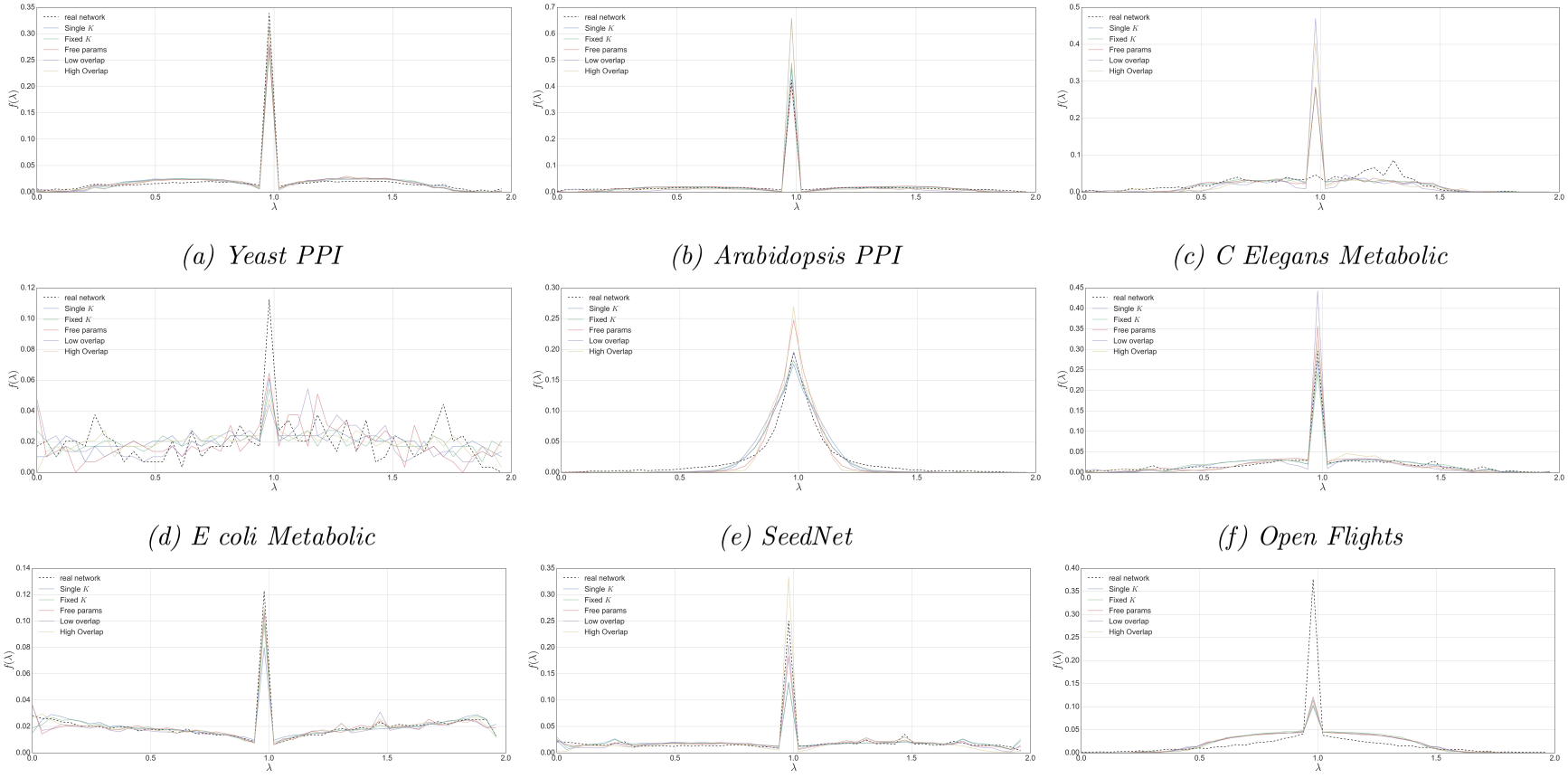


(a) Yeast PPI *(b) Arabidopsis PPI* *(c) C Elegans Metabolic*

(d) E coli Metabolic *(e) SeedNet* *(f) Open Flights*

(g) US Power Grid *(h) PGP* *(i) Hamster*

Figure B.7: Distribution of maximal modularity for best fit models. Histograms of 100 samples with kernel density estimates are shown. Colours indicate model fit for real world graphs (black dashes), single K (blue), fixed K (green), free parameters (red), low overlap (purple) and high overlap (yellow). Where dashes are not present, this is due to high levels of model inaccuracy.

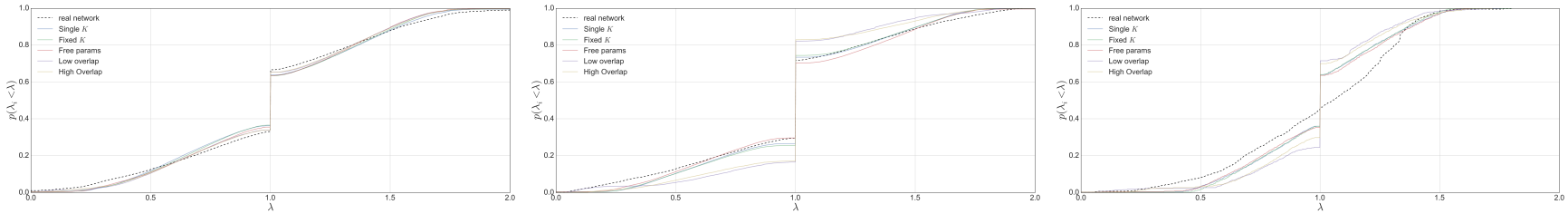


(a) Yeast PPI *(b) Arabidopsis PPI* *(c) C Elegans Metabolic*

(d) E coli Metabolic *(e) SeedNet* *(f) Open Flights*

(g) US Power Grid *(h) PGP* *(i) Hamster*

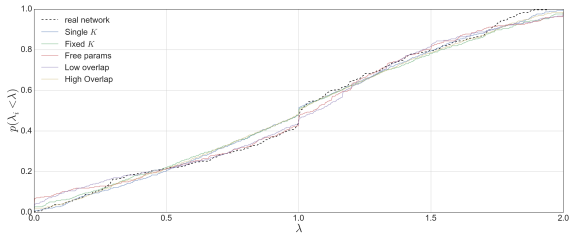
Figure B.8: Spectral distribution of networks. Histogram of eigenvalues of the normalised Laplacian. Colours indicate model fit for real world graphs (black dashes), single K (blue), fixed K (green), free parameters (red), low overlap (purple) and high overlap (yellow).



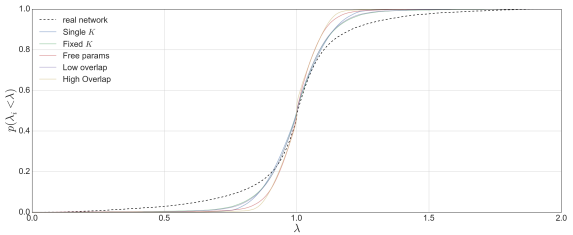
(a) Yeast PPI

(b) Arabidopsis PPI

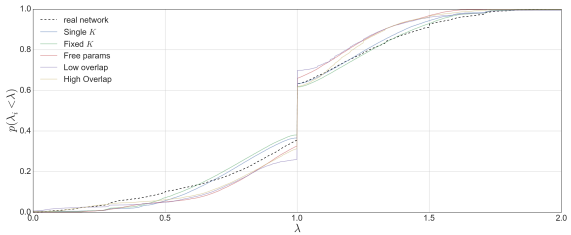
(c) C Elegans Metabolic



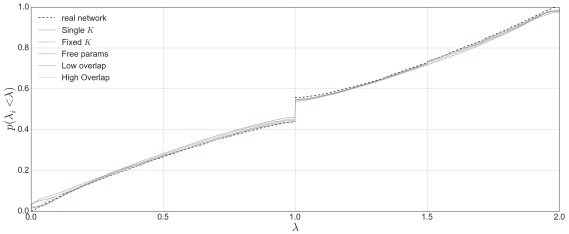
(d) E coli Metabolic



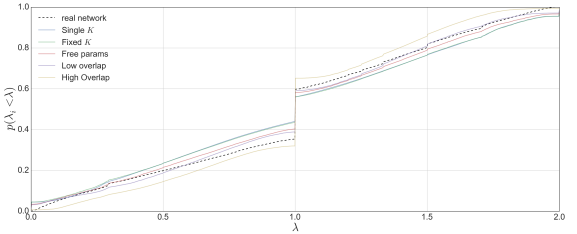
(e) SeedNet



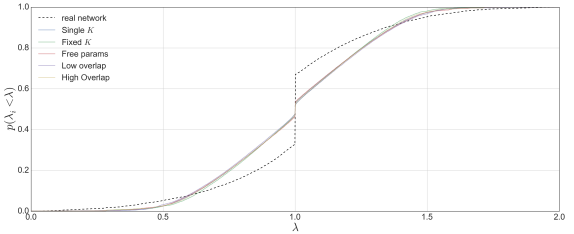
(f) Open Flights



(g) US Power Grid



(h) PGP



(i) Hamster

Figure B.9: Cumulative distribution of the eigenvalues of the Normalised Laplacian matrix. Colours indicate model fit for real world graphs (black dashes), single k (blue), fixed k (green), free parameters (red), low overlap (purple) and high overlap (yellow).

Appendix C

Benchmarking supplement

This appendix includes supplementary material for Chapter 6.

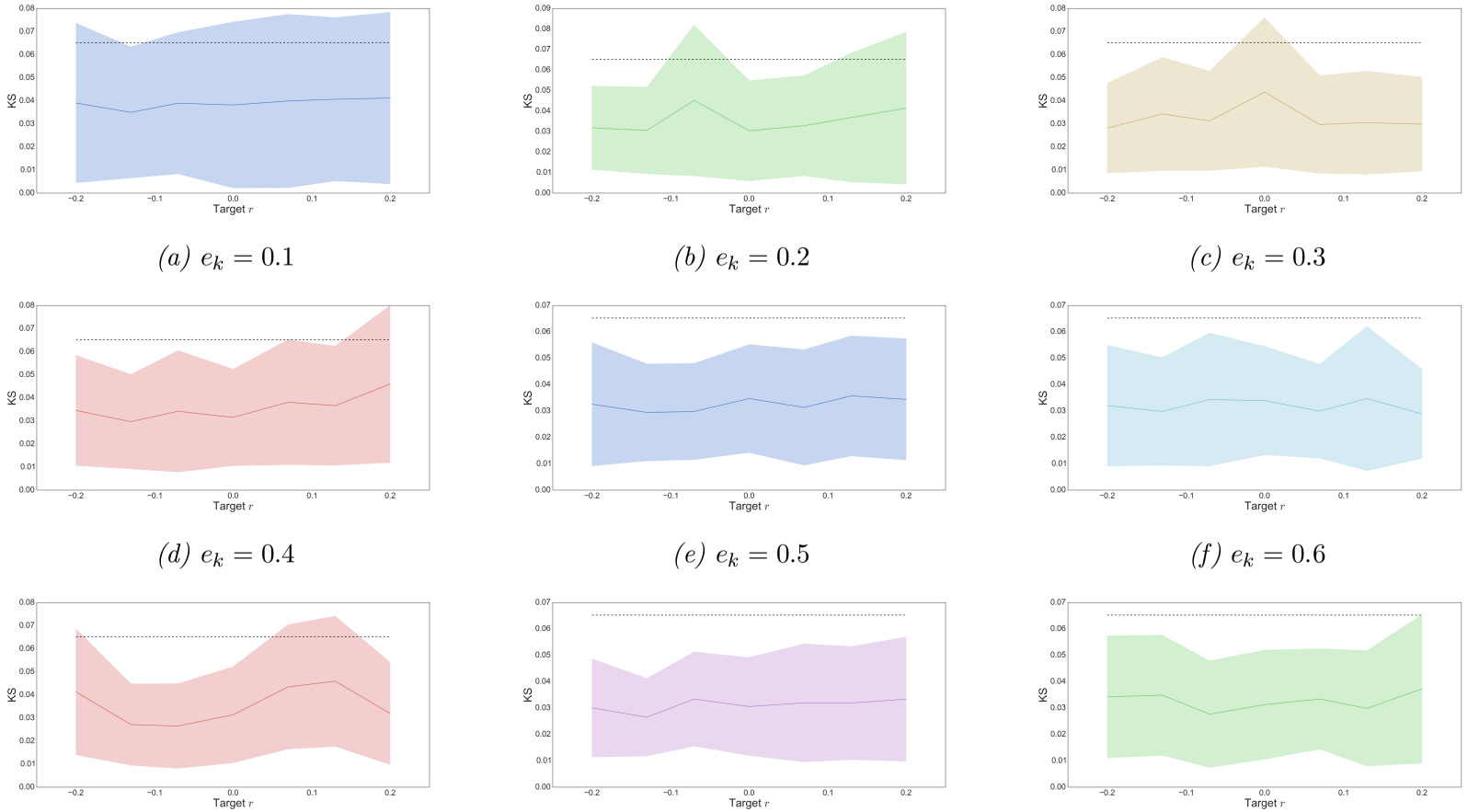
Figure C.1 shows the Kolmogorov-Smirnov distance between the target average cumulative degree distribution described in Section 6.4. These fits were achieved through the parameter selection based on particle swarm optimisation described in Section 5.2. Each plot relates to a fixed fraction of intra community edges, e_k between 0.1 and 0.9. The main line indicated the mean at increasing levels of target assortativity r between -0.2 and 0.2. The shaded area indicates the KS distance observed within two standard deviations from 100 realisations of CiGRAM with the selected parameters. The dashed line indicates two standard deviations of distance observed between the target model and degree distributions generated from 1000 realisations of CiGRAM. As the results show, most of the resulting degree distributions are within two standard deviations of the target model's expected KS distance, indicating good representation. The average CDF and CCDFs from 100 runs of the best fit model parameters are shown in Figures C.2 and C.3, respectively.

Figure C.4 shows the distributions for maximum degree observed in 100 realisations of the best fit parameters in the form of violin plots. On these plots the y axis shows one and two standard deviations, with a kernel density estimate of the distribution. Many of the distributions have a high level of variance but the resulting maximum degrees appear to be close to the target distribution shown in grey in all figures.

Figure C.5 shows violin plots for the level of assortativity generated by CiGRAM with the target parameters across ranges of e_k. The central dashed

line shows the target assortativity r. The violin plots show that the distributions for assortativity vary over a considerable range making exact fits difficult. As a consequence, in the tests for the normalised mutual information between detected and ground truth communities, the resulting graphs within CiGRAM are resampled until the target assortativity is within the shaded grey area of $r \pm 0.03$.

The clustering agreement Figures C.6 to C.8, shows the degree to which the algorithms tested in section 6.5.1 perform consistently. The darker the shade of red, the more consistently the two algorithms perform. Interestingly, for many of the networks in question, the results show no real sign of agreement. These results indicate the average normalised mutual information across the 32 replicate networks generated.

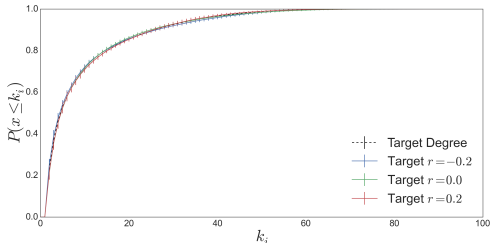


(a) $e_k = 0.1$ *(b)* $e_k = 0.2$ *(c)* $e_k = 0.3$

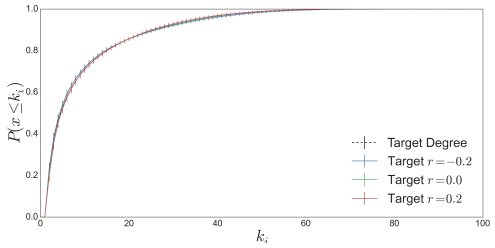
(d) $e_k = 0.4$ *(e)* $e_k = 0.5$ *(f)* $e_k = 0.6$

(g) $e_k = 0.7$ *(h)* $e_k = 0.8$ *(i)* $e_k = 0.9$

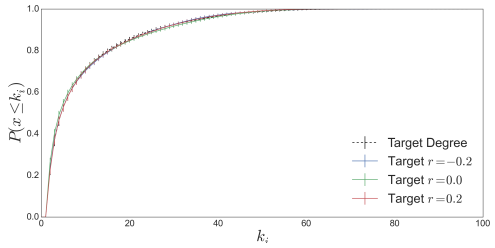
Figure C.1: Accuracy of best fit degree distributions by KS distance from target average CDF across range of e_k*. Black dotted line indicates two standard deviations of the target model degree distribution, shaded area indicates two standard deviations of the best fit model.*



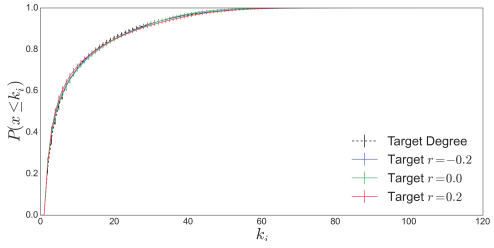
(a) $e_k = 0.1$



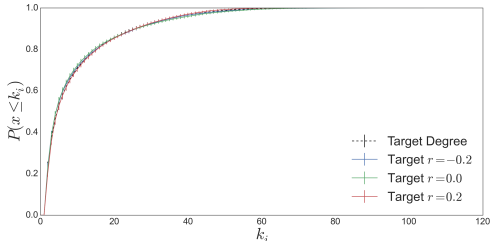
(b) $e_k = 0.2$



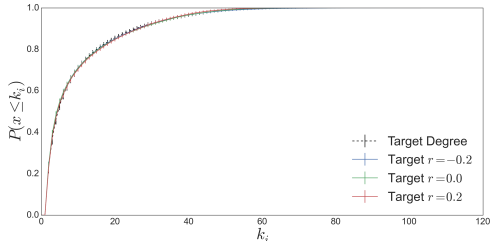
(c) $e_k = 0.3$



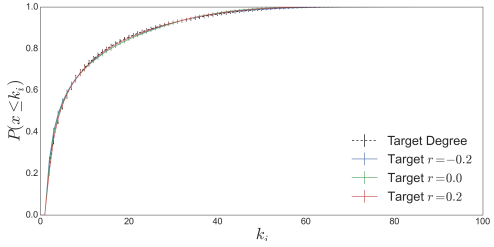
(d) $e_k = 0.4$



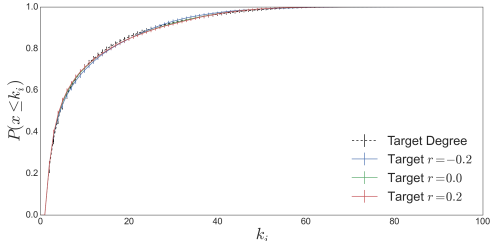
(e) $e_k = 0.5$



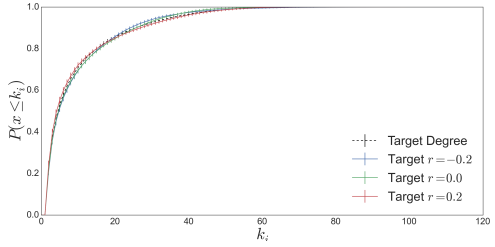
(f) $e_k = 0.6$



(g) $e_k = 0.7$

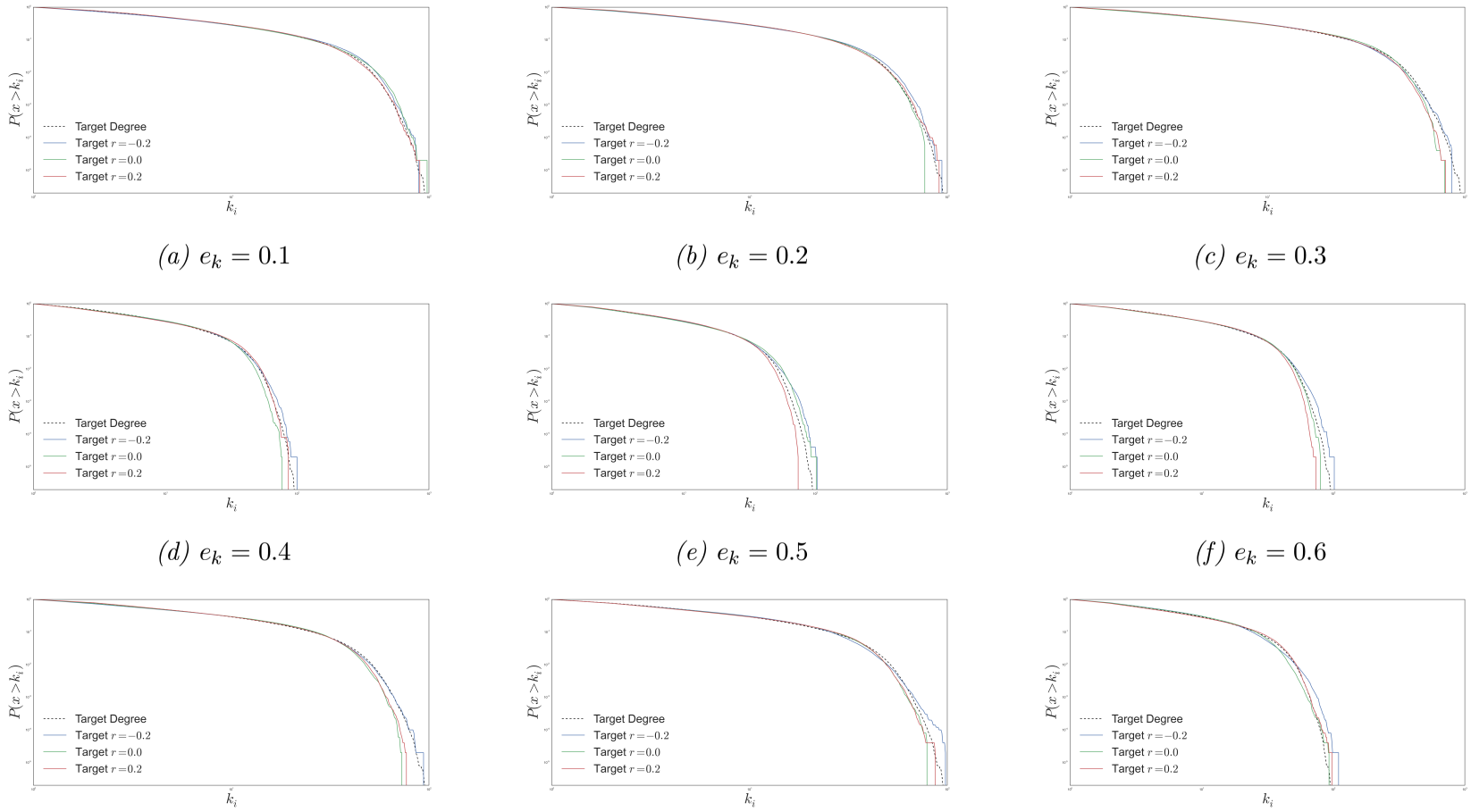


(h) $e_k = 0.8$



(i) $e_k = 0.9$

Figure C.2: Cumulative degree distribution plots for best fit assortative models at varying levels of e_k.

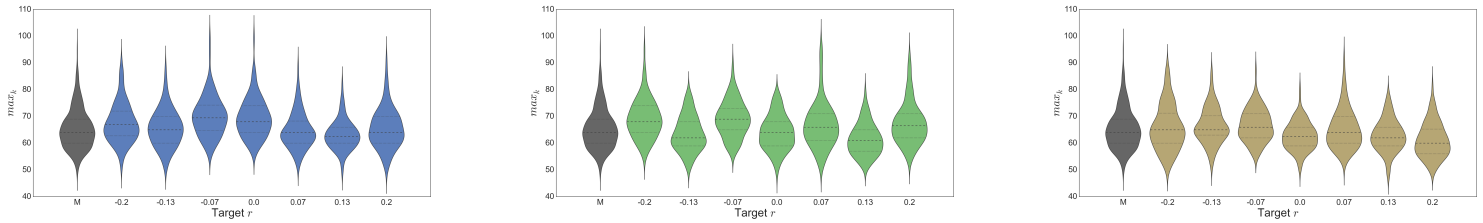


(a) $e_k = 0.1$ *(b)* $e_k = 0.2$ *(c)* $e_k = 0.3$

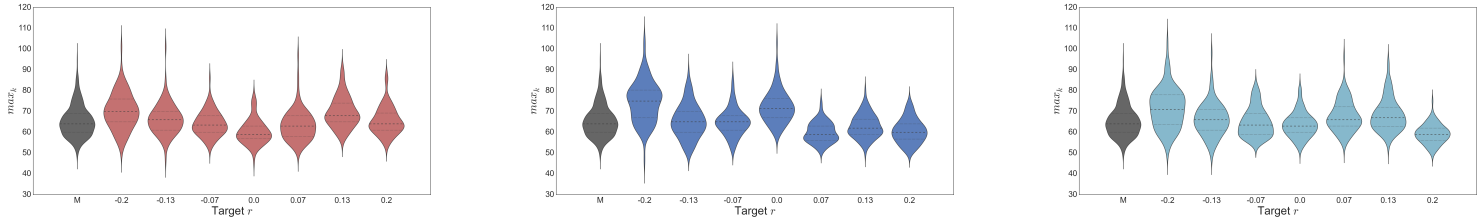
(d) $e_k = 0.4$ *(e)* $e_k = 0.5$ *(f)* $e_k = 0.6$

(g) $e_k = 0.7$ *(h)* $e_k = 0.8$ *(i)* $e_k = 0.9$

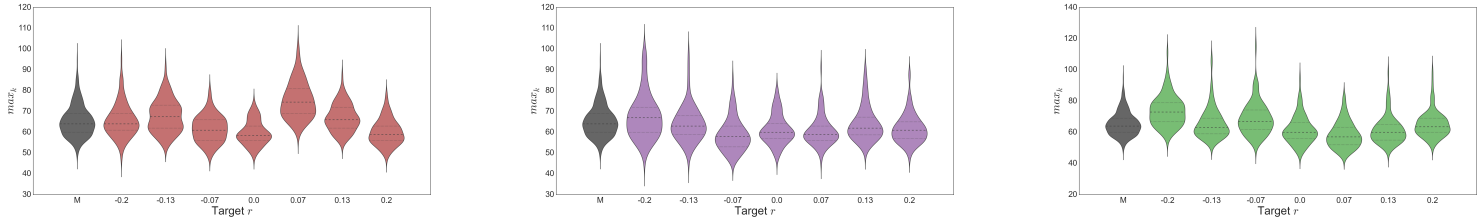
Figure C.3: Complementary cumulative degree distribution plots for best fit assortative models at varying levels of e_k*.*



(a) $e_k = 0.1$ *(b)* $e_k = 0.2$ *(c)* $e_k = 0.3$

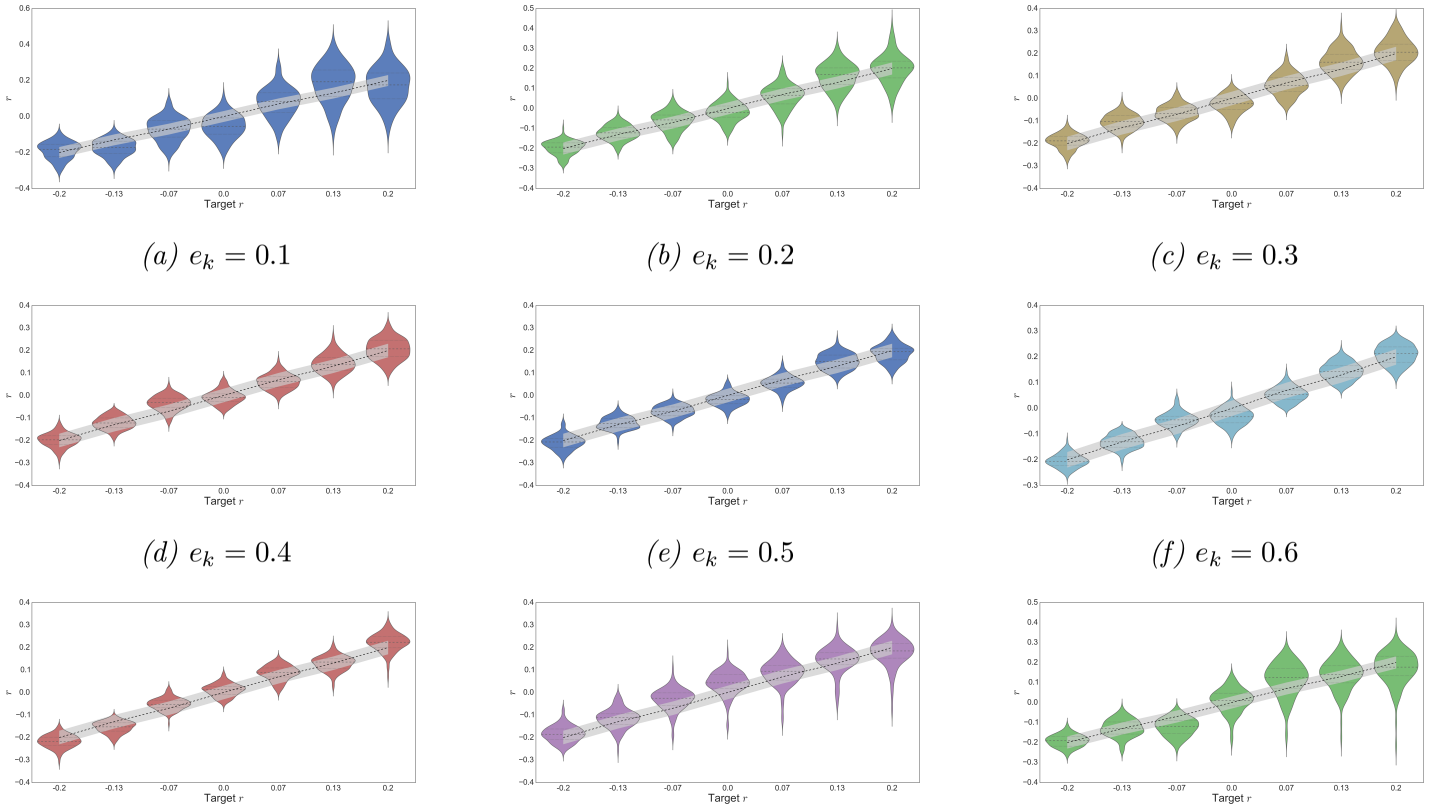


(d) $e_k = 0.4$ *(e)* $e_k = 0.5$ *(f)* $e_k = 0.6$



(g) $e_k = 0.7$ *(h)* $e_k = 0.8$ *(i)* $e_k = 0.9$

Figure C.4: Violin plots showing accuracy of maximum degree across range of e_k targets. Target r shown on the x axis. For comparison, the distribution of the target model maximum degree is shown in grey.

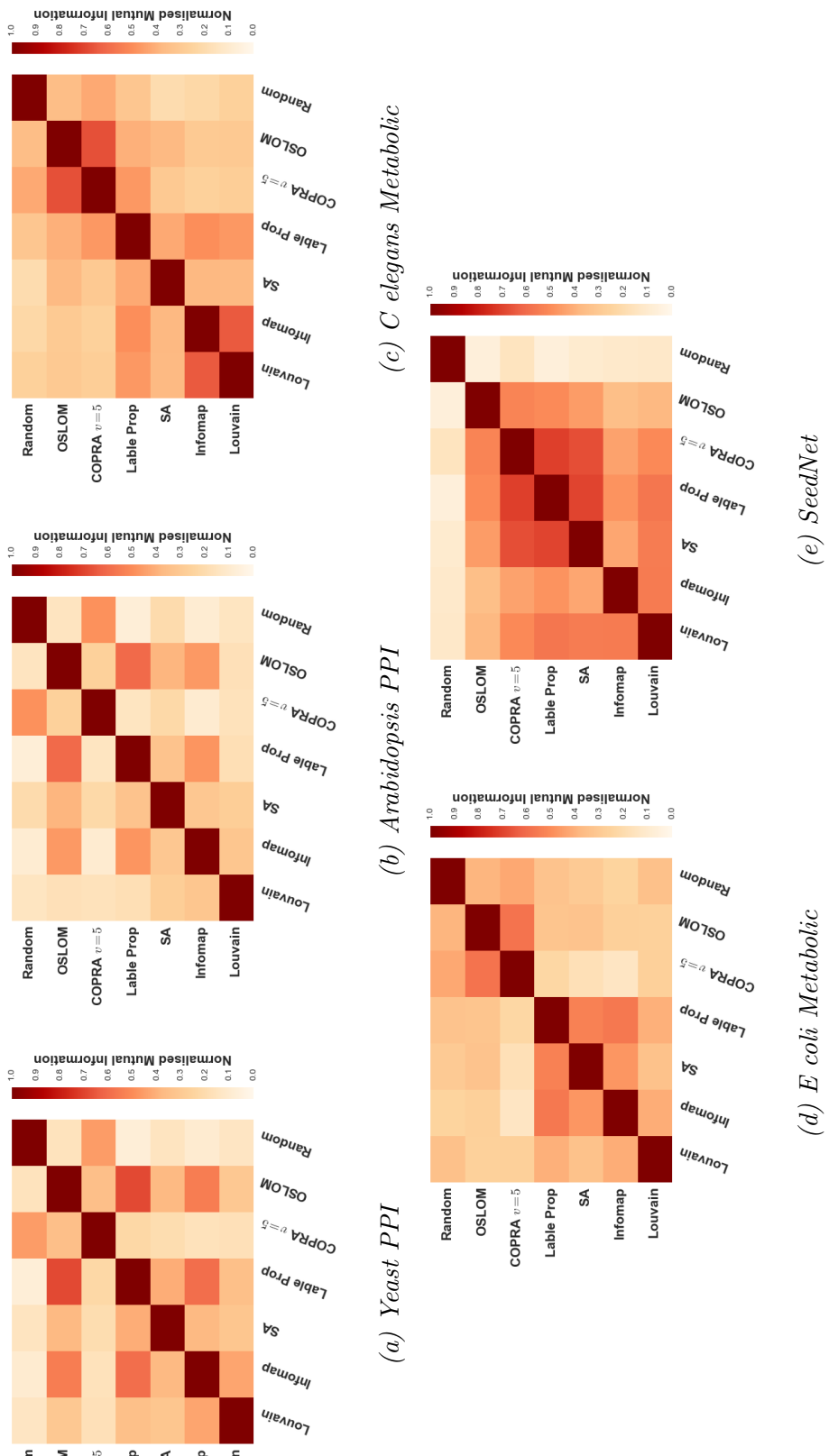


(a) $e_k = 0.1$ *(b)* $e_k = 0.2$ *(c)* $e_k = 0.3$

(d) $e_k = 0.4$ *(e)* $e_k = 0.5$ *(f)* $e_k = 0.6$

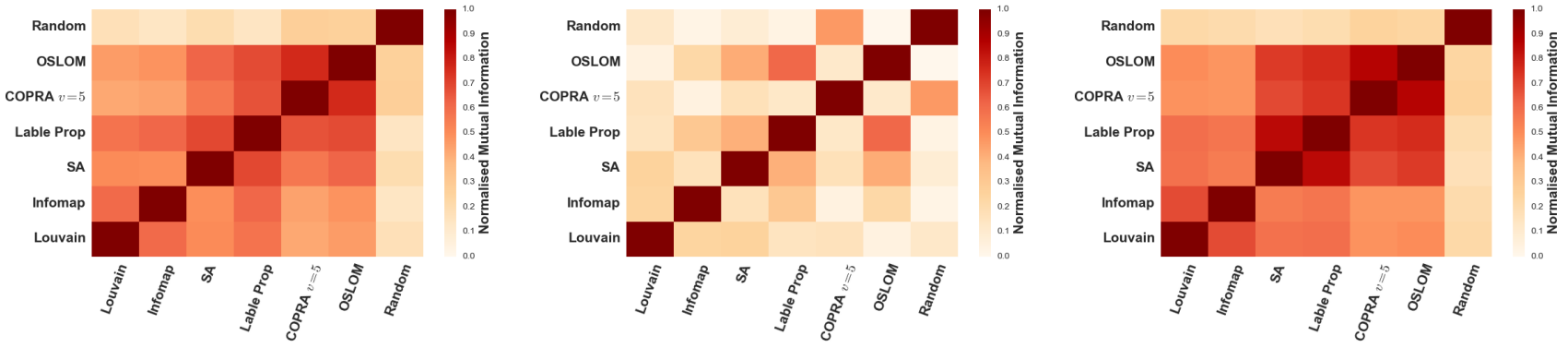
(g) $e_k = 0.7$ *(h)* $e_k = 0.8$ *(i)* $e_k = 0.9$

Figure C.5: Violin plots of assortativity for graphs generated with CiGRAM with best fit parameters. Each plot indicates the results for a fixed level of e_k*. Line indicates linear increase in target value shaded grey area indicates networks accepted in re-sampling for benchmarks. Results are shown for target values of* $r \in \{-0.2, -0.13, -0.07, 0.0, 0.07, 0.13, 0.2\}$

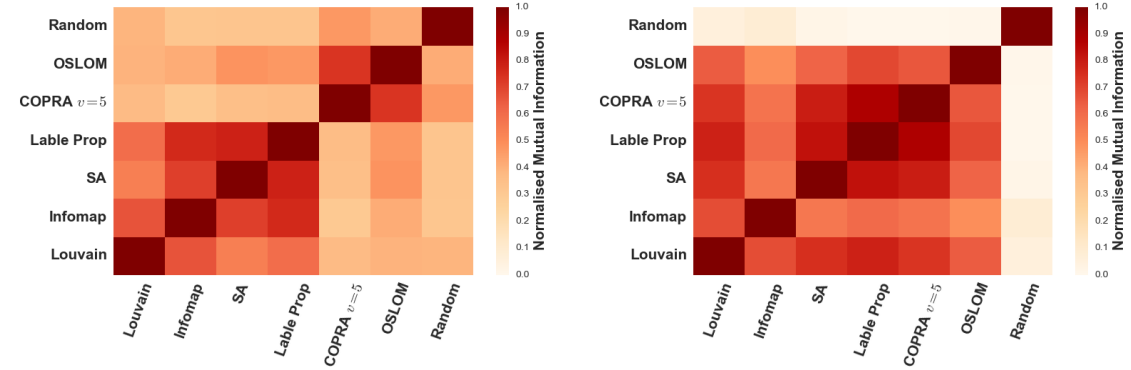


(a) Yeast PPI *(b) Arabidopsis PPI* *(c) C elegans Metabolic* *(d) E coli Metabolic* *(e) SeedNet*

Figure C.6: Normalised mutual information consensus matrix for agreement between algorithms on best fit biological networks with the Fixed K models.

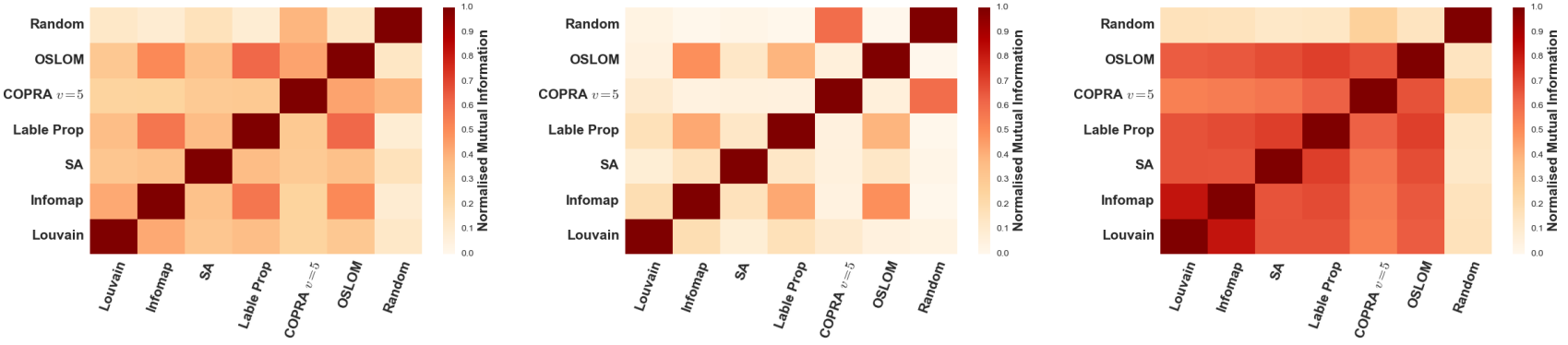


(a) Yeast PPI *(b) Arabidopsis PPI* *(c) C elegans Metabolic*

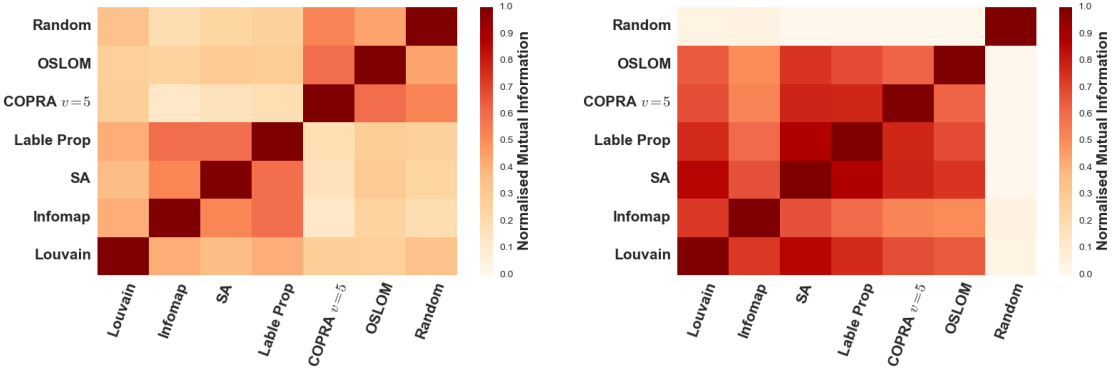


(d) E coli Metabolic *(e) SeedNet*

Figure C.7: Normalised mutual information consensus matrix for agreement between algorithms on best fit biological networks with the low Overlap models.



(a) Yeast PPI *(b) Arabidopsis PPI* *(c) C elegans Metabolic*



(d) E coli Metabolic *(e) SeedNet*

Figure C.8: Normalised mutual information consensus matrix for agreement between algorithms on best fit biological networks with the High Overlap models.

References

[1] D. Howe, M. Costanzo, P. Fey, T. Gojobori, L. Hannick, W. Hide, D. P. Hill, R. Kania, M. Schaeffer, S. St Pierre, S. Twigger, O. White, and S. Y. Rhe, "Big data: The future of biocuration," *Nature*, vol. 455, no. 7209, pp. 47–50, 2008.

[2] A.-L. Barabasi and Z. N. Oltvai, "Network biology: understanding the cell's functional organization," *Nature Reviews Genetics*, vol. 5, no. 2, pp. 101–113, 2004.

[3] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, "From molecular to modular cell biology," *Nature*, vol. 402, pp. C47–C52, 1999.

[4] M. Dreze, A.-R. Carvunis, B. Charloteaux, M. Galli, S. J. Pevzner, M. Tasan, Y.-Y. Ahn, P. Balumuri, A.-L. Barabási, V. Bautista, *et al.*, "Evidence for network evolution in an Arabidopsis interactome map," *Science*, vol. 333, no. 6042, pp. 601–607, 2011.

[5] M. E. J. Newman, "The structure and function of complex networks," *SIAM REVIEW*, vol. 45, pp. 167–256, 2003.

[6] Watts Duncan J. and Strogatz Steven H., "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, jun 1998.

[7] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[8] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[9] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, p. 046110, 2008.

[10] C. I. Del Genio, T. Gross, and K. E. Bassler, "All scale-free networks are sparse," *Physical Review Letters*, vol. 107, no. 17, p. 178701, 2011.

[11] L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. Villas Boas, "Characterization of complex networks: A survey of measurements," *Advances in Physics*, vol. 56, no. 1, pp. 167–242, 2007.

[12] A. Clauset, C. R. Shalizi, and M. E. Newman, "Power-law distributions in empirical data," *SIAM review*, vol. 51, no. 4, pp. 661–703, 2009.

[13] D. R. White and S. P. Borgatti, "Betweenness centrality measures for directed graphs," *Social Networks*, vol. 16, no. 4, pp. 335–346, 1994.

[14] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[15] W. E. Donath and A. J. Hoffman, "Lower bounds for the partitioning of graphs," *IBM Journal of Research and Development*, vol. 17, no. 5, pp. 420–425, 1973.

[16] D. Fay, A. W. Moore, K. Brown, M. Filosi, and G. Jurman, "Graph metrics as summary statistics for Approximate Bayesian Computation with application to network model parameter estimation," *Journal of Complex Networks*, p. cnu009, 2014.

[17] S. White and P. Smyth, "A spectral clustering approach to finding communities in graphs," in *SDM*, vol. 5, pp. 76–84, SIAM, 2005.

[18] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang, "Spectral redemption in clustering sparse networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110, no. 52, pp. 20935–20940, 2013.

[19] K. Mitra, A.-R. Carvunis, S. K. Ramesh, and T. Ideker, "Integrative approaches for finding modular structure in biological networks," *Nature Reviews Genetics*, vol. 14, no. 10, pp. 719–732, 2013.

[20] C. J. Stam and J. C. Reijneveld, "Graph theoretical analysis of complex networks in the brain," *Nonlinear biomedical physics*, vol. 1, no. 1, p. 3, 2007.

[21] T. Maniatis and R. Reed, "An extensive network of coupling among gene expression machines," *Nature*, vol. 416, no. 6880, pp. 499–506, 2002.

[22] B. H. Junker and F. Schreiber, *Analysis of biological networks*, vol. 2. John Wiley & Sons, 2011.

[23] T. Rolland, M. Taşan, B. Charloteaux, S. J. Pevzner, Q. Zhong, N. Sahni, S. Yi, I. Lemmens, C. Fontanillo, R. Mosca, *et al.*, "A Proteome-scale map of the human interactome network," *Cell*, vol. 159, no. 5, pp. 1212–1226, 2014.

[24] M.-S. Kim, S. M. Pinto, D. Getnet, R. S. Nirujogi, S. S. Manda, R. Chaerkady, A. K. Madugundu, D. S. Kelkar, R. Isserlin, S. Jain, *et al.*, "A draft map of the human proteome," *Nature*, vol. 509, no. 7502, pp. 575–581, 2014.

[25] J. K. Joung, E. I. Ramm, and C. O. Pabo, "A bacterial two-hybrid selection system for studying protein–DNA and protein–protein interactions," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 13, pp. 7382–7387, 2000.

[26] O. Puig, F. Caspary, G. Rigaut, B. Rutz, E. Bouveret, E. Bragado-Nilsson, M. Wilm, and B. Séraphin, "The tandem affinity purification (TAP) method: a general procedure of protein complex purification," *Methods*, vol. 24, no. 3, pp. 218–229, 2001.

[27] A. Brückner, C. Polge, N. Lentze, D. Auerbach, and U. Schlattner, "Yeast two-hybrid, a powerful tool for systems biology," *International journal of molecular sciences*, vol. 10, no. 6, pp. 2763–2788, 2009.

[28] C. Von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork, "Comparative assessment of large-scale data sets of protein–protein interactions," *Nature*, vol. 417, no. 6887, pp. 399–403, 2002.

[29] H. Huang and J. S. Bader, "Precision and recall estimates for two-hybrid screens," *Bioinformatics*, vol. 25, no. 3, pp. 372–378, 2009.

[30] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, *et al.*, "A comprehensive analysis of protein–protein interactions in Saccharomyces cerevisiae," *Nature*, vol. 403, no. 6770, pp. 623–627, 2000.

[31] A. Cherkasov, M. Hsing, R. Zoraghi, L. J. Foster, R. H. See, N. Stoynov, J. Jiang, S. Kaur, T. Lian, L. Jackson, *et al.*, "Mapping the protein interaction network in methicillin-resistant Staphylococcus aureus," *Journal of proteome research*, vol. 10, no. 3, pp. 1139–1150, 2011.

[32] S. V. Rajagopala, P. Sikorski, A. Kumar, R. Mosca, J. Vlasblom, R. Arnold, J. Franca-Koh, S. B. Pakala, S. Phanse, A. Ceol, A. Ceol, R. Häuser, G. Siszler, S. Wuchty, A. Emili, M. Babu, P. Aloy, R. Pieper, and P. Uetz, "The binary protein-protein interaction landscape of Escherichia coli," *Nature biotechnology*, vol. 32, no. 3, pp. 285–290, 2014.

[33] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguez, T. Doerks, M. Stark, J. Muller, P. Bork, L. J. Jensen, and C. von Mering, "The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored," *Nucleic acids research*, vol. 39, no. suppl 1, pp. D561–D568, 2011.

[34] A. Chatr-aryamontri, B.-J. Breitkreutz, S. Heinicke, L. Boucher, A. Winter, C. Stark, J. Nixon, L. Ramage, N. Kolas, L. O'Donnell, T. Reguly, A. Breitkreutz, A. Sellam, D. Chen, C. Chang, J. Rust, M. Livstone, R. Oughtred, K. Dolinski, and M. Tyers, "The BioGRID interaction database: 2013 update," *Nucleic acids research*, vol. 41, no. D1, pp. D816–D823, 2013.

[35] K. Venkatesan, J.-F. Rual, A. Vazquez, U. Stelzl, I. Lemmens, T. Hirozane-Kishikawa, T. Hao, M. Zenkner, X. Xin, K.-I. Goh, *et al.*, "An empirical framework for binary interactome mapping," *Nature methods*, vol. 6, no. 1, pp. 83–90, 2009.

[36] V. Spirin and L. A. Mirny, "Protein complexes and functional modules in molecular networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 21, pp. 12123–12128, 2003.

[37] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[38] S. Y. Rhee and M. Mutwil, "Towards revealing the functions of all genes in plants," *Trends in plant science*, vol. 19, no. 4, pp. 212–221, 2014.

[39] P. J. Schoonheim, M. P. Sinnige, J. A. Casaretto, H. Veiga, T. D. Bunney, R. S. Quatrano, and A. H. de Boer, "14-3-3 adaptor proteins are intermediates in ABA signal transduction during barley seed germination," *The Plant Journal*, vol. 49, no. 2, pp. 289–301, 2007.

[40] G. Gibson, "Microarray Analysis," *PLoS Biol*, vol. 1, p. e15, 10 2003.

[41] Z. Wang, M. Gerstein, and M. Snyder, "Rna-seq: a revolutionary tool for transcriptomics," *Nature Reviews Genetics*, vol. 10, no. 1, pp. 57–63, 2009.

[42] F. M. Giorgi, C. Del Fabbro, and F. Licausi, "Comparative study of RNA-seq-and Microarray-derived coexpression networks in Arabidopsis thaliana," *Bioinformatics*, 2013.

[43] G. W. Bassel, H. Lan, E. Glaab, D. J. Gibbs, T. Gerjets, N. Krasnogor, A. J. Bonner, M. J. Holdsworth, and N. J. Provart, "Genome-wide network model capturing seed germination reveals coordinated regulation of plant cellular phase transitions," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 23, pp. 9709–9714, 2011.

[44] B. Usadel, T. Obayashi, M. Mutwil, F. Giorgi, G. Bassel, M. Tanimoto, A. Chow, D. Steinhauser, S. Persson, and N. Provart, "Co-expression tools for plant biology: opportunities for hypothesis generation and caveats," *Plant, cell & environment*, vol. 32, no. 12, pp. 1633–1651, 2009.

[45] C. G. de Oliveira Dal'Molin, L.-E. Quek, R. W. Palfreyman, S. M. Brumbley, and L. K. Nielsen, "AraGEM, a genome-scale reconstruction of the primary metabolic network in Arabidopsis," *Plant physiology*, vol. 152, no. 2, pp. 579–589, 2010.

[46] A. Wagner and D. A. Fell, "The small world inside large metabolic networks," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 268, no. 1478, pp. 1803–1810, 2001.

[47] M. Kanehisa and S. Goto, "KEGG: Kyoto Encyclopedia of Genes and Genomes," *Nucleic Acids Research*, vol. 28, no. 1, pp. 27–30, 2000.

[48] T. Kelder, A. R. Pico, K. Hanspers, M. P. Van Iersel, C. Evelo, and B. R. Conklin, "Mining biological pathways using WikiPathways web services," *PloS one*, vol. 4, no. 7, p. e6447, 2009.

[49] I. M. Keseler, A. Mackie, M. Peralta-Gil, A. Santos-Zavaleta, S. Gama-Castro, C. Bonavides-Martínez, C. Fulcher, A. M. Huerta, A. Kothari, M. Krummenacker, M. Latendresse, L. Muñiz-Rascado, Q. Ong, S. Paley, I. Schröder, A. G. Shearer, P. Subhraveti, M. Traver, D. Weerasinghe, V. Weiss, J. Collado-Vides, R. P. Gunsalus, I. Paulsen, and P. D. Karp, "EcoCyc: fusing model organism databases with systems biology," *Nucleic acids research*, vol. 41, no. D1, pp. D605–D612, 2013.

[50] R. Caspi, T. Altman, R. Billington, K. Dreher, H. Foerster, C. A. Fulcher, T. A. Holland, I. M. Keseler, A. Kothari, A. Kubo, M. Krummenacker, M. Latendresse, L. A. Mueller, Q. Ong, S. Paley, P. Subhraveti, D. S. Weaver, D. Weerasinghe, P. Zhang, and P. D. Karp, "The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases," *Nucleic acids research*, vol. 42, no. D1, pp. D459–D471, 2014.

[51] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, pp. 651–654, 2000.

[52] E. Ravasz, A. Somera, D. Mongru, Z. Oltvai, and A. Barabási, "Hierarchical organization of modularity in metabolic networks," *Science*, vol. 297, no. 5586, pp. 1551–1555, 2002.

[53] S. Schuster, T. Dandekar, and D. A. Fell, "Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering," *Trends in biotechnology*, vol. 17, no. 2, pp. 53–60, 1999.

[54] R. Guimera and L. A. N. Amaral, "Functional cartography of complex metabolic networks," *Nature*, vol. 433, no. 7028, pp. 895–900, 2005.

[55] N. Gehlenborg, S. I. O'Donoghue, N. S. Baliga, A. Goesmann, M. A. Hibbs, H. Kitano, O. Kohlbacher, H. Neuweger, R. Schneider, D. Tenenbaum, *et al.*, "Visualization of omics data for systems biology," *Nature methods*, vol. 7, pp. S56–S68, 2010.

[56] G. A. Pavlopoulos, A.-L. Wegener, and R. Schneider, "A survey of visualization tools for biological network analysis," *BioData Min*, vol. 1, no. 1, p. 12, 2008.

[57] T. M. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Software: Practice and experience*, vol. 21, no. 11, pp. 1129–1164, 1991.

[58] S. Martin, W. Brown, R. Klavans, and K. Boyack, "OpenOrd: an open-source toolbox for large graph layout," in *IS&T/SPIE Electronic Imaging*, pp. 786806–786806, International Society for Optics and Photonics, 2011.

[59] V. Batagelj and A. Mrvar, "Pajek-program for large network analysis," *Connections*, vol. 21, no. 2, pp. 47–57, 1998.

[60] J. Köhler, J. Baumbach, J. Taubert, M. Specht, A. Skusa, A. Rüegg, C. Rawlings, P. Verrier, and S. Philippi, "Graph-based analysis and visu-

alization of experimental results with ONDEX," *Bioinformatics*, vol. 22, no. 11, pp. 1383–1390, 2006.

[61] P. Shannon, A. Markiel, O. Ozier, N. Baliga, J. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, "Cytoscape: a software environment for integrated models of biomolecular interaction networks," *Genome research*, vol. 13, no. 11, pp. 2498–2504, 2003.

[62] R. Saito, M. Smoot, K. Ono, J. Ruscheinski, P. Wang, S. Lotia, A. Pico, G. Bader, and T. Ideker, "A travel guide to Cytoscape plugins," *Nature Methods*, vol. 9, no. 11, pp. 1069–1076, 2012.

[63] S. Maere, K. Heymans, and M. Kuiper, "BiNGO: a Cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks," *Bioinformatics*, vol. 21, no. 16, pp. 3448–3449, 2005.

[64] M. Krull, S. Pistor, N. Voss, A. Kel, I. Reuter, D. Kronenberg, H. Michael, K. Schwarzer, A. Potapov, C. Choi, *et al.*, "TRANSPATH: an information resource for storing and visualizing signaling pathways and their pathological aberrations," *Nucleic acids research*, vol. 34, no. suppl 1, pp. D546–D551, 2006.

[65] L. A. Mueller, P. Zhang, and S. Y. Rhee, "AraCyc: a biochemical pathway database for Arabidopsis," *Plant Physiology*, vol. 132, no. 2, pp. 453–460, 2003.

[66] J. H. Morris, L. Apeltsin, A. M. Newman, J. Baumbach, T. Wittkop, G. Su, G. D. Bader, and T. E. Ferrin, "ClusterMaker: a multi-algorithm clustering plugin for Cytoscape," *BMC bioinformatics*, vol. 12, no. 1, p. 436, 2011.

[67] A. Lancichinetti, F. Radicchi, J. Ramasco, and S. Fortunato, "Finding statistically significant communities in networks," *PloS one*, vol. 6, no. 4, p. e18961, 2011.

[68] A. Bargiela and W. Pedrycz, *Granular computing: an introduction*. Springer Science & Business Media, 2003.

[69] J. Clune, J.-B. Mouret, and H. Lipson, "The evolutionary origins of modularity," *Proceedings of the Royal Society B: Biological Sciences*, vol. 280, no. 1755, 2013.

[70] R. Pastor-Satorras, E. Smith, and R. V. Solé, "Evolving protein interaction networks through gene duplication," *Journal of Theoretical biology*, vol. 222, no. 2, pp. 199–210, 2003.

[71] F. Chung, L. Lu, T. G. Dewey, and D. J. Galas, "Duplication models for biological networks," *Journal of computational biology*, vol. 10, no. 5, pp. 677–687, 2003.

[72] J. Hallinan, "Gene duplication and hierarchical modularity in intracellular interaction networks," *Biosystems*, vol. 74, no. 1, pp. 51–62, 2004.

[73] F. Emmert-Streib, "Limitations of gene duplication models: evolution of modules in protein interaction networks," *PloS one*, vol. 7, no. 4, p. e35531, 2012.

[74] J.-D. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, L. V. Zhang, D. Dupuy, A. J. Walhout, M. E. Cusick, F. P. Roth, and M. Vida, "Evidence for dynamically organized modularity in the yeast protein–protein interaction network," *Nature*, vol. 430, no. 6995, pp. 88–93, 2004.

[75] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, pp. 1118–1123, Jan. 2008.

[76] G. Palla, A.-L. Barabási, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.

[77] I. Vragović and E. Louis, "Network community structure and loop coefficient method," *Physical Review E*, vol. 74, no. 1, p. 016105, 2006.

[78] S. Gregory, "Fuzzy overlapping communities in networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2011, no. 02, p. P02017, 2011.

[79] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, p. 036106, 2007.

[80] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner, "On modularity clustering," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, no. 2, pp. 172–188, 2008.

[81] B. H. Good, Y.-A. de Montjoye, and A. Clauset, "Performance of modularity maximization in practical contexts," *Physical Review E*, vol. 81, no. 4, p. 046106, 2010.

[82] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[83] C. Pizzuti, "GA-Net: A genetic algorithm for community detection in social networks," in *Parallel Problem Solving from Nature–PPSN X*, pp. 1081–1090, Springer, 2008.

[84] M. Meilă, "Comparing clusterings by the variation of information," in *Learning theory and kernel machines*, pp. 173–187, Springer, 2003.

[85] P. Demartines and J. Hérault, "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets," *Neural Networks, IEEE Transactions on*, vol. 8, no. 1, pp. 148–154, 1997.

[86] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 1, pp. 36–41, 2007.

[87] J. P. Bagrow, "Communities and bottlenecks: Trees and treelike networks have high modularity," *Physical Review E*, vol. 85, no. 6, p. 066118, 2012.

[88] P. Zhang and C. Moore, "Scalable detection of statistically significant communities and hierarchies, using message passing for modularity," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 111, no. 51, pp. 18144–18149, 2014.

[89] A. Lancichinetti and S. Fortunato, "Consensus clustering in complex networks," *Scientific reports*, vol. 2, 2012.

[90] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[91] V. Černỳ, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of optimization theory and applications*, vol. 45, no. 1, pp. 41–51, 1985.

[92] E. Ziv, M. Middendorf, and C. H. Wiggins, "Information-theoretic approach to network modularity," *Physical Review E*, vol. 71, p. 046117, Apr 2005.

[93] A. J. Enright, S. Van Dongen, and C. A. Ouzounis, "An efficient algorithm for large-scale detection of protein families," *Nucleic acids research*, vol. 30, no. 7, pp. 1575–1584, 2002.

[94] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *Computer and Information Sciences-ISCIS 2005*, pp. 284–293, Springer, 2005.

[95] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[96] M. Rosvall and C. T. Bergstrom, "Multilevel Compression of Random Walks on Networks Reveals Hierarchical Organization in Large Integrated Systems," *PLoS ONE*, vol. 6, p. e18209, 04 2011.

[97] A. Lancichinetti, F. Radicchi, and J. J. Ramasco, "Statistical significance of communities in networks," *Physical Review E*, vol. 81, no. 4, p. 046110, 2010.

[98] S. Gregory, "Finding overlapping communities in networks by label propagation," *New Journal of Physics*, vol. 12, no. 10, p. 103018, 2010.

[99] P. Erdos and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hungar. Acad. Sci*, vol. 5, pp. 17–61, 1960.

[100] E. N. Gilbert, "Random graphs," *The Annals of Mathematical Statistics*, vol. 30, no. 4, pp. 1141–1144, 1959.

[101] M. Granovetter, "The strength of weak ties," *American journal of sociology*, vol. 78, no. 6, p. l, 1973.

[102] S. Vitali, J. B. Glattfelder, and S. Battiston, "The network of global corporate control," *PLoS ONE*, vol. 6, no. 10, p. e25995, 2011.

[103] S. Allesina and S. Tang, "Stability criteria for complex ecosystems," *Nature*, vol. 483, no. 7388, pp. 205–208, 2012.

[104] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[105] H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, no. 6833, pp. 41–42, 2001.

[106] M. P. H. Stumpf and M. A. Porter, "Critical Truths About Power Laws," *Science*, vol. 335, pp. 665–666, Feb. 2012.

[107] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical Review E*, vol. 72, no. 2, p. 027104, 2005.

[108] L. Li, D. Alderson, J. C. Doyle, and W. Willinger, "Towards a theory of scale-free graphs: Definition, properties, and implications," *Internet Mathematics*, vol. 2, no. 4, pp. 431–523, 2005.

[109] J.-F. Rual, K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G. F. Berriz, F. D. Gibbons, M. Dreze, N. Ayivi-Guedehoussou, *et al.*, "Towards a proteome-scale map of the human protein–protein interaction network," *Nature*, vol. 437, no. 7062, pp. 1173–1178, 2005.

[110] M. Vidal, M. E. Cusick, and A.-L. Barabasi, "Interactome networks and human disease," *Cell*, vol. 144, no. 6, pp. 986–998, 2011.

[111] R. Tanaka, T.-M. Yi, and J. Doyle, "Some protein interaction data do not exhibit power law statistics," *FEBS letters*, vol. 579, no. 23, pp. 5140–5144, 2005.

[112] T. Dobzhansky and T. G. Dobzhansky, *Genetics and the Origin of Species*. No. 11, Columbia University Press, 1937.

[113] S. Milgram, "The small world problem," *Psychology today*, vol. 2, no. 1, pp. 60–67, 1967.

[114] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[115] F. Chung and L. Lu, "The average distances in random graphs with given expected degrees," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 25, pp. 15879–15882, 2002.

[116] M. Molloy and B. Reed, "A critical point for random graphs with a given degree sequence," *Random structures & algorithms*, vol. 6, no. 2-3, pp. 161–180, 1995.

[117] J. Blitzstein and P. Diaconis, "A sequential importance sampling algorithm for generating random graphs with prescribed degrees," *Internet Mathematics*, vol. 6, no. 4, pp. 489–522, 2011.

[118] P. Erdos and T. Gallai, "Graphen mit punkten vorgeschriebenen grades," *Mat. Lapok*, vol. 11, pp. 264–274, 1960.

[119] M. E. Newman, "Assortative mixing in networks," *Physical review letters*, vol. 89, no. 20, p. 208701, 2002.

[120] O. Frank and D. Strauss, "Markov graphs," *Journal of the american Statistical association*, vol. 81, no. 395, pp. 832–842, 1986.

[121] S. Wasserman and P. Pattison, "Logit models and logistic regressions for social networks: I. an introduction to markov graphs and p*," *Psychometrika*, vol. 61, no. 3, pp. 401–425, 1996.

[122] T. A. Snijders, P. E. Pattison, G. L. Robins, and M. S. Handcock, "New specifications for exponential random graph models," *Sociological methodology*, vol. 36, no. 1, pp. 99–153, 2006.

[123] M. Brede and S. Sinha, "Assortative mixing by degree makes a network more unstable," *arXiv preprint cond-mat/0507710*, 2005.

[124] M. Boguná and R. Pastor-Satorras, "Epidemic spreading in correlated complex networks," *Physical Review E*, vol. 66, no. 4, p. 047104, 2002.

[125] Z. Tao, F. Zhongqian, and W. Binghong, "Epidemic dynamics on complex networks," *Progress in Natural Science*, vol. 16, no. 5, pp. 452–457, 2006.

[126] F. Papadopoulos, M. Kitsak, M. Á. Serrano, M. Boguná, and D. Krioukov, "Popularity versus similarity in growing networks," *Nature*, vol. 489, no. 7417, pp. 537–540, 2012.

[127] A. Quayle, A. Siddiqui, and S. Jones, "Modeling network growth with assortative mixing," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 50, no. 4, pp. 617–630, 2006.

[128] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Physical Review E*, vol. 80, no. 1, p. 016118, 2009.

[129] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, p. P09008, 2005.

[130] B. Karrer and M. E. Newman, "Stochastic blockmodels and community structure in networks," *Physical Review E*, vol. 83, no. 1, p. 016107, 2011.

[131] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Physical review E*, vol. 68, no. 6, p. 065103, 2003.

[132] A. F. McDaid, D. Greene, and N. Hurley, "Normalized mutual information to evaluate overlapping community finding algorithms," *arXiv preprint arXiv:1110.2515*, 2011.

[133] D. Hric, R. K. Darst, and S. Fortunato, "Community detection in networks: Structural communities versus ground truth," *Physical Review E*, vol. 90, no. 6, p. 062805, 2014.

[134] C. Seshadhri, T. G. Kolda, and A. Pinar, "Community structure and scale-free collections of Erdős-Rényi graphs," *Physical Review E*, vol. 85, no. 5, p. 056109, 2012.

[135] P. Lamesch, T. Z. Berardini, D. Li, D. Swarbreck, C. Wilks, R. Sasidharan, R. Muller, K. Dreher, D. L. Alexander, M. Garcia-Hernandez, A. S. Karthikeyan, C. H. Lee, W. D. Nelson, L. Ploetz, S. Singh, A. Wensel, and H. Eva, "The Arabidopsis Information Resource (TAIR): improved gene annotation and new tools," *Nucleic acids research*, vol. 40, no. D1, pp. D1202–D1210, 2012.

[136] G. D. Bader and C. W. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC bioinformatics*, vol. 4, no. 1, p. 2, 2003.

[137] T. Narayanan and S. Subramaniam, "Community Structure Analysis of Gene Interaction Networks in Duchenne Muscular Dystrophy," *PLOS ONE*, vol. 8, no. 6, p. e67237, 2013.

[138] S. Treviño III, Y. Sun, T. F. Cooper, and K. E. Bassler, "Robust detection of hierarchical communities from Escherichia coli gene expression data," *PLoS computational biology*, vol. 8, no. 2, p. e1002391, 2012.

[139] C.-C. Lin, C.-H. Lee, C.-S. Fuh, H.-F. Juan, and H.-C. Huang, "Link Clustering Reveals Structural Characteristics and Biological Contexts in Signed Molecular Networks," *PloS one*, vol. 8, no. 6, p. e67089, 2013.

[140] J. Handl, J. Knowles, and D. B. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics*, vol. 21, no. 15, pp. 3201–3212, 2005.

[141] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. S. others, "Gene Ontology: tool for the unification of biology," *Nature genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[142] M. Quint, H.-G. Drost, A. Gabel, K. K. Ullrich, M. Bönn, and I. Grosse, "A transcriptomic hourglass in plant embryogenesis," *Nature*, 2012.

[143] S. Zhong, Z. Fei, Y.-R. Chen, Y. Zheng, M. Huang, J. Vrebalov, R. McQuinn, N. Gapper, B. Liu, J. Xiang, Y. Shao, and J. J. Giovannoni, "Single-base resolution methylomes of tomato fruit development reveal epigenome modifications associated with ripening," *Nature biotechnology*, vol. 31, no. 2, pp. 154–159, 2013.

[144] M. Fujisawa, T. Nakano, Y. Shima, and Y. Ito, "A large-scale identification of direct targets of the tomato MADS box transcription factor RIPENING INHIBITOR reveals the regulation of fruit ripening," *The Plant Cell Online*, vol. 25, no. 2, pp. 371–386, 2013.

[145] M. Bemer, R. Karlova, A. R. Ballester, Y. M. Tikunov, A. G. Bovy, M. Wolters-Arts, P. de Barros Rossetto, G. C. Angenent, and R. A. de Maagd, "The tomato FRUITFULL homologs TDR4/FUL1 and MBP7/FUL2 regulate ethylene-independent aspects of fruit ripening," *The Plant Cell Online*, vol. 24, no. 11, pp. 4437–4451, 2012.

[146] R. Karlova, F. M. Rosin, J. Busscher-Lange, V. Parapunova, P. T. Do, A. R. Fernie, P. D. Fraser, C. Baxter, G. C. Angenent, and R. A. de Maagd, "Transcriptome and metabolite profiling show that APETALA2a is a major regulator of tomato fruit ripening," *The Plant Cell Online*, vol. 23, no. 3, pp. 923–941, 2011.

[147] B. J. W. Dekkers, S. Pearce, R. P. van Bolderen-Veldkamp, A. Marshall, P. Widera, J. Gilbert, H.-G. Drost, G. W. Bassel, K. Müller, J. R. King, A. T. A. Wood, I. Grosse, M. Quint, N. Krasnogor, G. Leubner-Metzger, M. J. Holdsworth, and L. Bentsink, "Transcriptional dynamics of two seed compartments with opposing roles in Arabidopsis seed germination," *Plant Physiol.*, vol. 163, pp. 205–15, Sept. 2013.

[148] G. Seymour, C. Hodgman, J. P. Gilbert, P. Widera, and N. Krasnogor, "FruitNet: A resource for investigiating fruit ripening," *to be submitted*, 2015.

[149] A. D. Perkins and M. A. Langston, "Threshold selection in gene co-expression networks using spectral graph theory techniques," *BMC bioinformatics*, vol. 10, no. Suppl 11, p. S4, 2009.

[150] R. Finkelstein, W. Reeves, T. Ariizumi, and C. Steber, "Molecular Aspects of Seed Dormancy," *Plant Biology*, vol. 59, no. 1, p. 387, 2008.

[151] S. Penfield, Y. Li, A. D. Gilday, S. Graham, and I. A. Graham, "Arabidopsis ABA INSENSITIVE4 regulates lipid mobilization in the embryo and reveals repression of seed germination by the endosperm," *The Plant Cell Online*, vol. 18, no. 8, pp. 1887–1899, 2006.

[152] M. Ogawa, A. Hanada, Y. Yamauchi, A. Kuwahara, Y. Kamiya, and S. Yamaguchi, "Gibberellin biosynthesis and response during Arabidopsis seed germination," *The Plant Cell Online*, vol. 15, no. 7, pp. 1591–1604, 2003.

[153] K. Nakabayashi, M. Okamoto, T. Koshiba, Y. Kamiya, and E. Nambara, "Genome-wide profiling of stored mRNA in Arabidopsis thaliana seed germination: epigenetic and genetic regulation of transcription in seed," *The Plant Journal*, vol. 41, no. 5, pp. 697–709, 2005.

[154] E. Carrera, T. Holman, A. Medhurst, D. Dietrich, S. Footitt, F. L. Theodoulou, and M. J. Holdsworth, "Seed after-ripening is a discrete developmental pathway associated with specific gene networks in Arabidopsis," *The Plant Journal*, vol. 53, no. 2, pp. 214–224, 2008.

[155] G. W. Bassel, P. Fung, T.-f. F. Chow, J. A. Foong, N. J. Provart, and S. R. Cutler, "Elucidating the germination transcriptional program using small molecules," *Plant physiology*, vol. 147, no. 1, pp. 143–155, 2008.

[156] Y. Yamauchi, M. Ogawa, A. Kuwahara, A. Hanada, Y. Kamiya, and S. Yamaguchi, "Activation of gibberellin biosynthesis and response pathways by low temperature during imbibition of Arabidopsis thaliana seeds," *The Plant Cell Online*, vol. 16, no. 2, pp. 367–378, 2004.

[157] C. S. Cadman, P. E. Toorop, H. W. Hilhorst, and W. E. Finch-Savage, "Gene expression profiles of Arabidopsis Cvi seeds during dormancy cycling

indicate a common underlying dormancy control mechanism," *The Plant Journal*, vol. 46, no. 5, pp. 805–822, 2006.

[158] B. Zhang and S. Horvath, "A general framework for weighted gene co-expression network analysis," *Statistical applications in genetics and molecular biology*, vol. 4, no. 1, 2005.

[159] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," *Physical Review E*, vol. 80, no. 5, p. 056117, 2009.

[160] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, no. 3, p. 033015, 2009.

[161] B. Karrer, E. Levina, and M. E. Newman, "Robustness of community structure in networks," *Physical Review E*, vol. 77, no. 4, p. 046119, 2008.

[162] S. Carbon, A. Ireland, C. J. Mungall, S. Shu, B. Marshall, S. Lewis, *et al.*, "AmiGO: online access to ontology and annotation data," *Bioinformatics*, vol. 25, no. 2, pp. 288–289, 2009.

[163] D. W. Huang, B. T. Sherman, Q. Tan, J. Kir, D. Liu, D. Bryant, Y. Guo, R. Stephens, M. W. Baseler, H. C. Lane, and R. A. Lempicki, "DAVID Bioinformatics Resources: expanded annotation database and novel algorithms to better extract biology from large gene lists," *Nucleic acids research*, vol. 35, no. suppl 2, pp. W169–W175, 2007.

[164] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists," *Nucleic acids research*, vol. 37, no. 1, pp. 1–13, 2009.

[165] P. Khatri and S. Drăghici, "Ontological analysis of gene expression data: current tools, limitations, and open problems," *Bioinformatics*, vol. 21, no. 18, pp. 3587–3595, 2005.

[166] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 289–300, 1995.

[167] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of molecular biology*, vol. 215, no. 3, pp. 403–410, 1990.

[168] G. Bassel, E. Glaab, J. Marquez, M. Holdsworth, and J. Bacardit, "Functional network construction in Arabidopsis using rule-based machine learning on large-scale data sets," *The Plant Cell Online*, vol. 23, no. 9, pp. 3101–3116, 2011.

[169] J. Gillis and P. Pavlidis, ""Guilt by association" is the exception rather than the rule in gene networks," *PLoS computational biology*, vol. 8, no. 3, p. e1002444, 2012.

[170] W. Aiello, F. Chung, and L. Lu, "A random graph model for power law graphs," *Experimental Mathematics*, vol. 10, no. 1, pp. 53–66, 2001.

[171] S. Horvath and J. Dong, "Geometric interpretation of gene coexpression network analysis," *PLoS computational biology*, vol. 4, no. 8, p. e1000117, 2008.

[172] M. Penrose, *Random geometric graphs*, vol. 5. Oxford University Press Oxford, 2003.

[173] M. Á. Serrano, M. Boguñá, and F. Sagués, "Uncovering the hidden geometry behind metabolic networks," *Molecular BioSystems*, vol. 8, no. 3, pp. 843–850, 2012.

[174] K. V. Mardia and P. E. Jupp, *Directional statistics*, vol. 494. Wiley. com, 2009.

[175] P. S. Efraimidis and P. G. Spirakis, "Weighted random sampling with a reservoir," *Information Processing Letters*, vol. 97, no. 5, pp. 181–185, 2006.

[176] A. Singhal, "Modern information retrieval: A brief overview," *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35–43, 2001.

[177] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social networks*, vol. 5, no. 2, pp. 109–137, 1983.

[178] J. Chen and B. Yuan, "Detecting functional modules in the yeast protein–protein interaction network," *Bioinformatics*, vol. 22, no. 18, pp. 2283–2290, 2006.

[179] D. A. Schult and P. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conferences (SciPy 2008)*, vol. 2008, pp. 11–16, 2008.

[180] A. Sanfeliu and K.-S. Fu, "A distance measure between attributed relational graphs for pattern recognition," *Systems, Man and Cybernetics, IEEE Transactions on*, no. 3, pp. 353–362, 1983.

[181] G. Jurman, R. Visintainer, and C. Furlanello, "An introduction to spectral distances in networks," in *Neural Nets WIRN10: Proceedings of the 20th Italian Workshop on Neural Nets*, vol. 226, p. 227, IOS Press, 2011.

[182] B. Pincombe, "Detecting changes in time series of network graphs using minimum mean squared error and cumulative summation," *ANZIAM Journal*, vol. 48, pp. 450–473, 2007.

[183] A. Banerjee, "Structural distance and evolutionary relationship of networks," *Biosystems*, vol. 107, no. 3, pp. 186–196, 2012.

[184] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Nov. 1995.

[185] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[186] K. Parsopoulos and M. Vrahatis, "Particle swarm optimizer in noisy and continuously changing environments," *methods*, vol. 5, no. 6, p. 23, 2001.

[187] Y.-j. Gong and J. Zhang, "Small-world particle swarm optimization with topology adaptation," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pp. 25–32, ACM, 2013.

[188] D. Ekman, S. Light, Å. K. Björklund, and A. Elofsson, "What properties characterize the hub proteins of the protein-protein interaction network

of Saccharomyces cerevisiae?," *Genome biology*, vol. 7, no. 6, p. R45, 2006.

[189] T. Opsahl, F. Agneessens, and J. Skvoretz, "Node Centrality in Weighted Networks: Generalizing Degree and Shortest Paths," *Social Networks*, vol. 3, no. 32, pp. 245–251, 2010.

[190] J. Kunegis, "KONECT – The Koblenz Network Collection," in *Proc. Int. Conf. on World Wide Web Companion*, pp. 1343–1350, 2013.

[191] M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, and A. Arenas, "Models of Social Networks based on Social Distance Attachment," *Physical Review E*, vol. 70, no. 5, p. 056122, 2004.

[192] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 1990.

[193] T. Schank and D. Wagner, "Approximating clustering coefficient and transitivity," *Journal of Graph Algorithms and Applications*, vol. 9, 2005.

[194] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 181–213, 2015.

[195] G. K. Orman, V. Labatut, and H. Cherifi, "Towards realistic artificial benchmark for community detection algorithms evaluation," *International Journal of Web Based Communities*, vol. 9, no. 3, pp. 349–370, 2013.

[196] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 43, 2013.

[197] J. Besag, "Statistical analysis of non-lattice data," *The statistician*, pp. 179–195, 1975.

[198] Ö. N. Yaveroğlu, N. Malod-Dognin, D. Davis, Z. Levnajic, V. Janjic, R. Karapandza, A. Stojmirovic, and N. Pržulj, "Revealing the Hidden Language of Complex Networks," *Scientific reports*, vol. 4, 2014.

[199] M. Sales-Pardo, R. Guimera, A. A. Moreira, and L. A. N. Amaral, "Extracting the hierarchical organization of complex systems," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 39, pp. 15224–15229, 2007.

[200] L. A. Mueller, T. H. Solow, N. Taylor, B. Skwarecki, R. Buels, J. Binns, C. Lin, M. H. Wright, R. Ahrens, Y. Wang, E. V. Herbst, E. R. Keyder, N. Menda, D. Zamir, and S. D. Tanksley, "The SOL Genomics Network. A comparative resource for Solanaceae biology and beyond," *Plant physiology*, vol. 138, no. 3, pp. 1310–1317, 2005.

[201] C. T. Lopes, M. Franz, F. Kazi, S. L. Donaldson, Q. Morris, and G. D. Bader, "Cytoscape Web: an interactive web-based network browser," *Bioinformatics*, vol. 26, no. 18, pp. 2347–2348, 2010.

[202] J. Taubert, K. Hassani-Pak, N. Castells-Brooke, and C. J. Rawlings, "Ondex Web: web-based visualization and exploration of heterogeneous biological networks," *Bioinformatics*, vol. 30, no. 7, pp. 1034–1035, 2014.